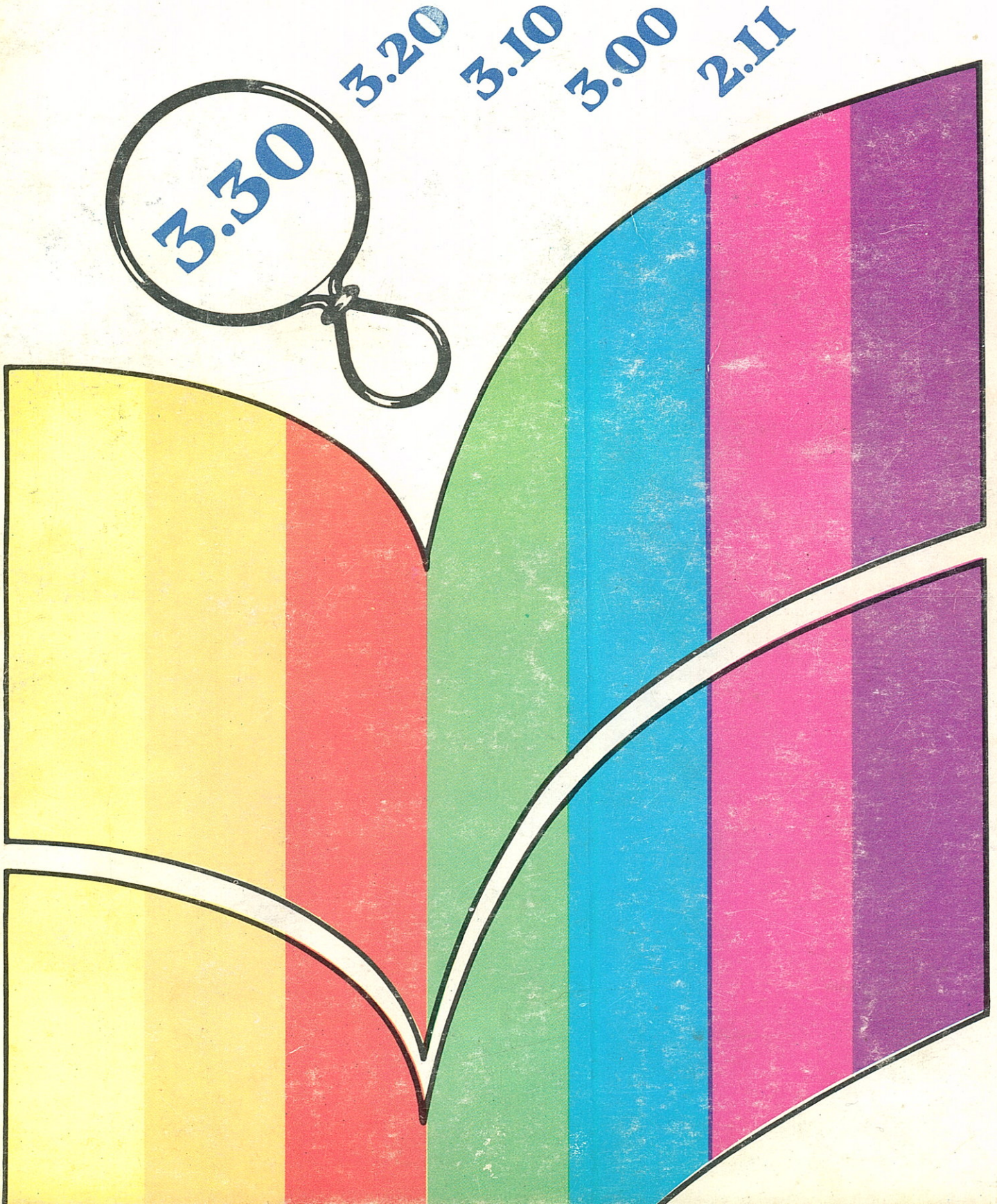
IBM PC DOS

I. A PC DOS HASZNÁLATA



} megszakításvektor lekérdezése
ES:BX-be

} elmentés

} megszakításvektor
felülírása

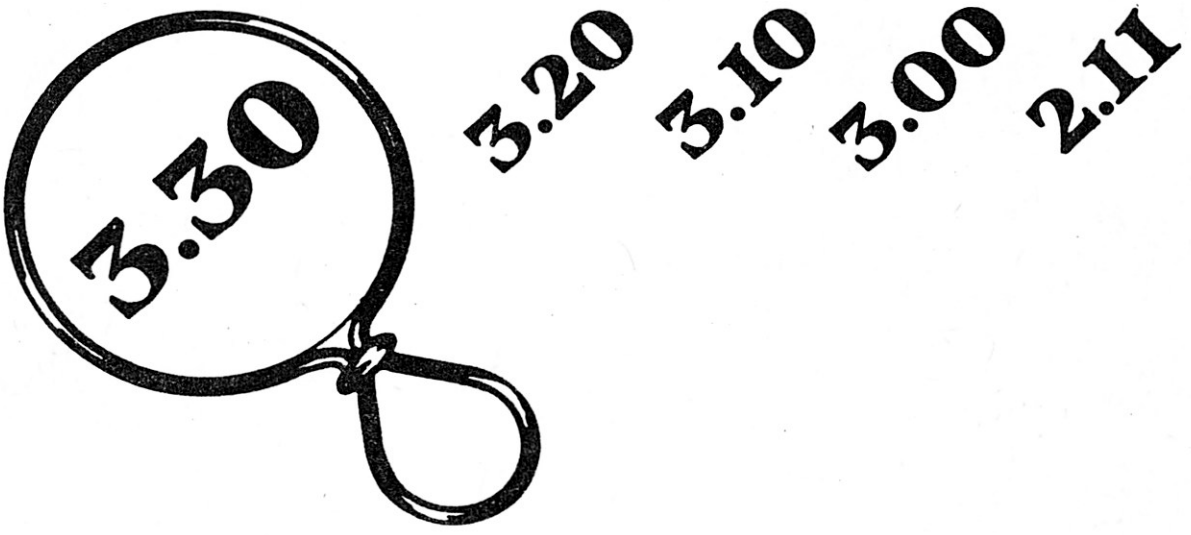
} programbefejezés
rezidens módon.

DEL billentyű

ALT
↑
ALT 11110111

IBM PC DOS

I. A PC DOS HASZNÁLATA



ALKALMAZÁSTECHNIKAI TANÁCSADÓ SZOLGÁLAT
BUDAPEST, 1988

Szerkesztette: Szenes Katalin I-II.
Dr. Úry László III.

Készítette:		
	Bernáth István	1, 2, 8 és 9. fejezet
	Darvas László	3, 4, 5, 6, 7, 11, 12, 13, 14, 15, 16 és 17. fejezet
	Orosz Judit	10. fejezet
	Seres József	3, 4, 5, 6, 7, 11, 12, 13, 14, 15, 16, és 17. fejezet
	Úry László	3.xx kiegészítések

III. kötet: Orosz Judit
Baki Zoltán
Pajor Gábor

Lektorálta: Éltető László
Szenes Katalin
Darvas László
Donát János

Munkatársak: Szegedi Gyöngyi
Kökényesi Judit

Kiadó: LSI Alkalmazástechnikai Tanácsadó Szolgálat
Felelős kiadó: dr. Kovács Magda
Témafelelős: Székely László
Szedés: LSI ATSz Info Osztály
Technikai szekesztő: Nagy Olivér
Engedélyszám: 57152
ISBN összkötet: 963 592 814 9
ISBN I. kötet: 963 592 811 4
Készült a Szabadság MgTsz Nyomdaüzemében, Gyál 88-19/sz.
Felelős vezető: Tóth Antal

Előszó

Az IBM PC számítógépekre készült PC DOS operációs rendszer filozófiája híven tükrözi, hogy személyi számítógépekre (PC - Personal Computer) készült, azaz a gép előtt ülő, azt interaktívan kezelni akaró programozó számára. Lehetőségeit a felhasználó elsősorban közvetlenül kiadható parancsokkal éri el. A rendszert és az IBM PC assembler-t mélyebben ismerők számára pedig úgynevezett DOS funkciók állnak rendelkezésre, a software feladatok megvalósítása érdekében.

Ennyiben nagyon hasonlít egyik elődjéhez, a CP/M - szintén mikroszámítógépes - operációs rendszerhez, de annál funkció-állománya jóval bővebb, nagy interrupt-készlethez biztosít hozzáférést és ami a legfontosabb, a file-okat bonyolult fa-hierarchiákba lehet rendezni, ez pedig nemcsak a felhasználók, hanem az egyes feladatok file-halmazainak kényelmes, némi biztonságot is adó strukturálását is lehetővé teszi.

Könyvünket egyaránt szánjuk az IBM PC-vel most ismerkedő (bár lehetőleg egy kevés számítástechnikai alapismerettel már rendelkező), és a már tapasztalt programozóknak és rendszerprogramozóknak.

A DOS egyes változatai új verziószámot kapnak, amikor a rendszerek lényeges új vonásokkal egészülnek ki. Ezek közül a 2.10, 3.00, 3.10, 3.20 és a 3.30 verziókkal foglalkozunk.

A DOS indítása, file-, fix (Winchester) és cserélhető lemezekre vonatkozó parancsainak ismertetése után bemutatjuk a parancsok file-okban való tárolását és a parancs-file-ok indítását.

A PC DOS változatokkal az EDLIN nevű sorszerkesztő program áll a rendszerrel egyidőben a felhasználó rendelkezésére. Ennek parancsaival részletesen foglalkozunk.

Assembler programozáshoz szükséges nyomkövető-hibakereső program a DEBUG.

Futtatható programot a programszerkesztővel (konszolidátornak, linkernek is nevezik más gépeken) állíthatunk elő, ez a LINK program. Ez lehetővé teszi eredetileg különféle forrásnyelven és assemblerben készült programok összeépítését, és programkönyvtár file-ok felhasználását is.

A kezdők számára - a DEBUG-ról szólók kivételével - első lépésként ezen információk elsajátítását ajánljuk.

A fentiek után a rendszer konfigurálásával, a DOS speciális lehetőségeivel, belső felépítésével, a fix lemezen létrehozható fa-strukturájú file-szerkezettel, az operációs rendszer munkaterületeivel és vezérlő blokkjaival foglalkozunk.

Részletesen tárgyaljuk a DOS különböző verzióinál rendelkezésre álló funkciókat és azokat az interrupt-okat (megszakításokat), amelyekhez a felhasználó is hozzáfér, természetesen assembler szinten.

Egyes DOS parancsok ismertetésénél kitérünk a megfelelő PROPOS operációs rendszer szempontjaira is.

A függelék a rendszer hibaüzenetein kívül hasznos listákat (parancsok, alparancsok) és táblázatokat (file-név kiterjesztés, eszköznevek) tartalmaz.

Az index (tárgymutató) a fontosabb fogalmak főbb és magyarázó lelőhelyeit közli.

Az irodalomjegyzékben megadjuk azokat a könyveket, amelyek információit saját tapasztalatunkon kívül itt felhasználtuk.

Természetesen nem oldható meg az összes könyv teljes tudásanyagának közreadása egy ilyen kézikönyvben, amely egy kicsit tankönyv is, de reméljük, hogy a legtöbb, még rendszerprogramozási szintű feladat megoldásához is elég segítséget adunk.

Az alábbiakban felsoroljuk az LSI ATSz egyéb, IBM PC alkalmazóknak szóló kiadványait. A könyvek megrendelhetők az LSI ATSz címén. (Budapest 1300, Pf. 114.)

Megjelent:

Erdős Iván – Rusznák László: **IBM PC felhasználói és technikai információs kártya**

Dr. Pajor Gábor: **IBM PC-ről kezdő felhasználóknak I. A hardver; II. A szoftver**

Kápolnai A. – Geszti P.: **IBM PC BASIC fejlesztőrendszer**

Lipovszki György: **IBM PC FORTH**

Ráth György: **Programvédelmi rendszerek IBM PC-re**

Pirkó József: **3D perspektívikus grafika Turbo Pascal-ban IBM PC-n**

Dr. Úry László: **SYMPHONY I.**

Dr. Barakonyi Károly: **Táblázatkezelő rendszerek IBM PC-re**
(Lotus 1-2-3, Framework, Calcstar)

REXLIB
(Képernyőkezelő, file-kezelő rezidens rutin- és utility-csomag programozói kézikönyve Turbo Pascal programnyelven fejlesztők számára MS-DOS és NOVELL hálózati környezetben. Ára: 205.-Ft)

Előkészületben:

Dr. Úry László: **SYMPHONY II.**
(Várható megjelenés: 1988. december)

Pirkó József: **Turbo Pascal kezdőknek** (4.0 verzióig IBM PC-re)
(Várható megjelenés: 1989. I. negyedév)

Bóér - Dóra - Fenyő - Seres: **Az IBM PC belső felépítése**
(Várható megjelenés: 1989. I. negyedév)

Mondovics János – Soltész Erzsébet: **dBASE III PLUS, CLIPPER**
(Várható megjelenés: 1989. I. negyedév)

Szenes Katalin

TARTALOMJEGYZÉK

I. kötet

Előszó 3
Tartalomjegyzék 5

1. FEJEZET
Bevezetés 11
1.1 Mi a PC DOS? 11
1.2 A PC DOS futtatásának hardware feltételei 13
1.3 A PC DOS verziói 15

2. FEJEZET
PC DOS alapismeretek 17

2.1 A PC DOS funkcionális részei 18
2.2 A PC DOS lemez-kezelése 18
2.3 A PC DOS indítása 21
2.3.1 A DOS elindítása 21
2.3.2 Ismételt indítás (restart) 23
2.3.3 Egyéb tennivalók 24
2.3.4 Fix lemezegység használata 25
2.4 A billentyűzet kezelése, DOS parancsok kiadása 27
2.5 Lemez és file kezelés 33
2.5.1 A fastruktúrájú tartalomjegyzék 33
2.5.2 File- és tartalomjegyzék nevek 35
2.5.3 Aktuális lemezegység 38
2.5.4 Keresési sorrend 38
2.6. A parancsok szintaxisa és a leírásuknál alkalmazott jelölésrendszer 39

3. FEJEZET
DOS: file-kezelő parancsok 41

3.1 COPY - másolás 41
3.2 XCOPY - file-ok csoportos másolása 45
3.3 REPLACE - file-ok cseréje 47
3.4 COMP - összehasonlítás 48
3.5 ERASE (DEL) - törlés 51
3.6 RENAME - átnevezés 52
3.7 RECOVER - helyreállítás 53
3.8 ATTRIB - file attributúmainak beállítása 54
3.9 SHARE - file megosztási környezet kezelése 55
3.10 TYPE - kiíratás 56
3.11 PRINT - kinyomtatás 57
3.12 FASTOPEN - file-ok elérésének gyorsítása 59

4. FEJEZET

DOS: lemezkezelési parancsok 60

4.1 MKDIR - altartalomjegyzék létrehozása 60
4.2 CHDIR - aktuális tartalomjegyzék kijelölése 61
4.3 RMDIR - altartalomjegyzék törlése 62
4.4 TREE - tartalomjegyzék struktúrájának kiíratása 63
4.5 PATH - bejárási utak kijelölése programokra 63
4.6 FORMAT - lemez formázása 64
4.7 SELECT - új rendszerlemez előállítása adott billentyűzet rutinnal, dátum és idő formátummal 68
4.8 SYS - rendszer-file-ok másolása 69
4.9 DIR - tartalomjegyzék kiíratása 70
4.10 VOL - kötetazonosító kiíratása 71
4.11 LABEL kötetazonosító megadása 72
4.12 DISKCOPY - lemez másolása 72
4.13 DISKCOMP - lemezek összehasonlítása 74
4.14 CHKDSK - lemez vizsgálata 75
4.15 VERIFY - írás utáni ellenőrzés 77
4.16 SUBST tartalomjegyzék helyettesítése lemezegység betűjelével 77
4.17 JOIN - lemezegység és tartalomjegyzék összekapcsolása 78
4.18 APPEND - bejárási utak kijelölése tetszőleges file-hoz 79

5. FEJEZET

A fix lemez kezelése 81

5.1 FDISK - fix lemez kialakítása 81
5.1.1 DOS-partíciók eiőállítása 82
5.1.2 Áktív partíció változtatása 84
5.1.3 DOS-partíció törlése 84
5.1.4 A partíció adatok kiíratása 85
5.1.5 A következő fix lemez kiválasztása 85
5.2 Fix lemez archiválása 85
5.2.1 BACKUP - file-ok archiválása 85
5.2.2 RESTORE - file-ok visszatöltése 88

6. FEJEZET

A rendszer működését befolyásoló parancsok és kapcsolók 90
6.1 COMMAND - újabb parancsfeldolgozó program elindítása 90
6.2 EXIT - visszatérés az elsődleges parancsfeldolgozóhoz 91
6.3 CLS - képernyő törlése 92
6.4 DATE - dátum megadása 92
6.5 TIME - idő megadása 93
6.6 VER - DOS verzió lekérdezése 94
6.7 ASSIGN - periféria-hozzárendelések megváltoztatása 94

6.8 BREAK - a < Break > billentyű figyelése 95
6.9 MODE - perifériák működésmódjának beállítása 96
6.10 KEYBxx - nemzeti billentyűzet kiválasztása 101
6.11 GRAFTABL - grafikai jelek betöltése 102
6.12 GRAPHICS - grafikus nyomtatás lehetősége 103

7. FEJEZET
A parancs-file-ok 105

7.1 A parancs-file-ok meghívása és paraméterezése 105
7.1.1 Formális paraméterek 106
7.1.2 Környezeti változók használata 106
7.2 Az AUTOEXEC.BAT file 107
7.3 DOS alparancsok 108
7.3.1 REM - megjegyzés 108
7.3.2 ECHO - parancsok megjelenítése 108
7.3.3 PAUSE - feldolgozás megszakítása 109
7.3.4 IF - feltételes végrehajtás 110
7.3.5 GOTO - ugrás a parancs-file-ban 112
7.3.6 FOR - DOS parancsok ismételt végrehajtása 112
7.3.7 SHIFT - a paraméter léptetése 113
7.3.8 CALL - parancs-file hívása 114

8. FEJEZET
A sorszerkesztő program (EDLIN) 115

8.1 Az EDLIN indítása 116
8.2 Az EDLIN parancsai 117
8.2.1 I (insert) - sorok beszúrása 119
8.2.2 L (list) - sorok listázása 120
8.2.3 P (page) - sorok listázása laponként 122
8.2.4 Sorszám - sor szerkesztése 123
8.2.5 D (delete) - sorok törlése 125
8.2.6 E (end) - szerkesztés befejezése tárolással 128
8.2.7 Q (quit) -kilépés a szerkesztésből 128
8.2.8 M (move) - sorok áthelyezése 129
8.2.9 C (copy) - sorok másolása 131
8.2.10 S (search) - karaktersorozat keresése 132
8.2.11 R (replace) - karaktersorozat helyettesítése 135
8.2.12 T (transfer) - sorok beszúrása lemezről 137
8.2.13 W (write) - sorok kiírása lemezre 138
8.2.14 A (append) - sorok hozzáfűzése lemezről 139

9. FEJEZET

A nyomkövető-hibakereső program (DEBUG) 141

9.1 A DEBUG indítása 142
9.2 A DEBUG parancsai 143
9.2.1 D (dump) - memóriatartalom kiíratása 145
9.2.2 R (register) - regiszterek kiíratása, megváltoztatása 146
9.2.3 F (fill) - memória feltöltése 147
9.2.4 E (enter) - memóriatartalom megváltoztatása 148
9.2.5 C (compare) - memóriaterületek összehasonlítása 149
9.2.6 M (move) - memóriaterület átmásolása 150
9.2.7 S (search) - keresés a memóriában 150
9.2.8 Q (quit) - kilépés a DEBUG-ból 151
9.2.9 A (assemble) - assemblálás 151
9.2.10 U (unassemble) - disassemblálás 153
9.2.11 G (go) - program futtatása 154
9.2.12 T (trace) - lépésenkénti végrehajtás 155
9.2.13 P - utasítássorozat végrehajtása 155
9.2.14 N (name) - a file kijelölése 156
9.2.15 L (load) - file (kijelölt szektorok) betöltése 157
9.2.16 W (write) - file (kijelölt szektorok) kiírása 158
9.2.17 H - hexadecimális műveletvégzés 160
9.2.18 I - input a kijelölt port-ról 160
9.2.19 O - output a kijelölt port-ra 160

10. FEJEZET

Futtatható program előállítása - programszerkesztés 161

10.1 A LINK indítása 161
10.1.1 Programszerkesztés PC DOS parancs-sorral 161
10.1.2 Programszerkesztés interaktív üzemmódban 162
10.1.3 Programszerkesztés előre összeállított válaszfile-lal 164
10.1.4 Könyvtárak elérési útjainak megadása 165
10.1.5 A map-file szerkezete 166
10.1.6 Ideiglenes lemez-file - VM.TMP 167
10.2 A LINK opciói 167
10.2.1 Megszakítás a szerkesztés alatt - PAUSE 167
10.2.2 Public szimbólumok táblájának előállítása 168
10.2.3 A verem-méret meghatározása - STACK 168
10.2.4 A kezdőcím maximumra állítása - HIGH 169
10.2.5 Helyfoglalás az adatcsoportnak - DSALLOCATION 169
10.2 6 Forrás-sorok kezdőcímének kiírása - LINE 170
10.2.7 Szegmensszám beállítása - X 170
10.2.8 'Régi' fordítók használata 170

11. FEJEZET

A rendszer konfigurálása 171

11.1 A konfigurációs file kezelése 171
11.2 A konfigurációs parancsok 171
11.2.1 BREAK - a Break billentyű figyelése 171
11.2.2 BUFFERS - lemez-pufferek száma 172
11.2.3 COUNTRY - dátum, idő és egyéb formátum megadása 173
11.2.4 DEVICE - periféria-kezelő rutinok 174
- A kibővített képernyő- és billentyűzetkezelés (ANSI.SYS) 177
- DISPLAY.SYS - kód-lapok használatának engedélyezése a képernyőn 178
- A DIVER.SYS használata 179
- PRINTER.SYS - kód-lapok használatának engedélyezése a nyomtatóra 180
- VDISK.SYS - memórialemez használata 181

11.2.5 FCBS - maximálisan megnyitható FCB-k száma 182
11.2.6 FILES - egyidejűleg megnyitható file-o száma 182
11.2.7 LASTDRIVE - a legnagyobb lemezegység megadása 182
11.2.8 SHELL - felhasználói parancs-processzor 183
11.2.9 STACKS - a DOS veremhasználatának módosítása 183

12. FEJEZET

A PC DOS speciális lehetőségei 184

12.1 A standard I/O átirányítása 184
12.2 A standard I/O összekapcsolása 185
12.3 Speciális DOS parancsok 186
12.3.1 MORE - képernyő oldalakra tagolása 186
12.3.2 FIND - keresés, válogatás 186
12.3.3 SORT - rendezés 187
12.3.4 PROMPT - a prompt karakter megváltoztatása 189
12.3.5 CTTY - alternatív konzol kijelölése 190
12.3.6 EXE2BIN - átalakítás EXE-ről COM formátumra 191
12.3.7 SET - programkörnyezet kezelése 191
12.3.8 NLSFUNC - ország-függő információk kezelése 193
12.3.9 CHCP - kódlap-váltás valamennyi eszközre 193

II. kötet

13. FEJEZET

A PC DOS belső struktúrája

14. FEJEZET

Lemezkezelés és a felhasználói perifériakezelők szerkezete

15. FEJEZET
A DOS memóriakezelése, munkaterületei és vezérlőblokkjai

16. FEJEZET
DOS megszakítások

17. FEJEZET
DOS funkciók

FÜGGELÉKEK
A. A DOS üzenetei
B. Parancsok, alparancsok csoportosított betűrendes listája
C. Parancsok, alparancsok formája
D. File-név kiterjesztések
E. Eszköznevek listája
Tárgymutató
Irodalomjegyzék

1. Bevezetés

Ebben a fejezetben megismerkedünk a PC DOS operációs rendszer eredetével és fontosabb jellemzőivel, majd megvizsgáljuk, milyen hardware-rel rendelkező számítógép szükséges a PC DOS futtatásához. Végül áttekintjük a PC DOS elterjedtebb verzióinak (változatainak) specialitásait.

1.1 Mi a PC DOS?

Mielőtt rátérnénk a PC DOS leírására, tekintsük át röviden, mi a mikroszámítógépeknél az operációs rendszerek jelentősége, feladata. Az operációs rendszer maga is egy programrendszer, melynek programjai közvetítő szerepet töltenek be a felhasználó (és programjai), valamint a számítógép hardware erőforrásai között. Ezek a programok - többek között - a következő alapvető feladatokat látják el:

- programok betöltése a tárba és elindítása,
- input - output műveletek végzése (billentyűzet olvasása, kiírás a képernyőre ill. nyomtatóra, lemezek írása és olvasása, lemezes adatállományok létrehozása, másolása, törlése stb.),
- a lemezek tartalomjegyzékének (directory-jának) létrehozása, karbantartása és listázása,
- a billentyűzetről kiadott parancsok értelmezése és végrehajtása,
- a működés közben fellépett hibák kezelése.

Az a mód és sebesség, ahogyan az operációs rendszer e feladatokat végrehajtja, valamint a rendszer egyéb szolgáltatásainak bősége jelentősen befolyásolják számítógépünk viselkedését, meghatározzák azt a környezetet, amelyben a géppel dolgozhatunk. Az operációs rendszer minősége így gyakran fontosabb a felhasználó számára, mint a számítógép hardware tulajdonságai.

Az operációs rendszerrel akkor kerülünk közvetlen kapcsolatba, amikor a billentyűzetről parancsokat adunk ki, és azokat a rendszer végrehajtja (például file másolása, program elindítása).

Ne feledkezzünk meg azonban arról sem, hogy az általunk írt, vagy készen vásárolt programok is futásuk során számtalanszor igénybe veszik az operációs rendszer szolgáltatásait (többek között az input-output műveletek végrehajtásakor). Ebből következik, hogy a közvetlenül futtatható (gépi kódban írt, illetve arra lefordított) programok nemcsak hardware, hanem oprációs rendszer-függők is; működésük általában csak egy adott operációs rendszer keretein belül lehetséges.

Az egyszerűbb mikrogépekbe gyárilag beépítik (beégetik) az operációs rendszert. Nagyobb teljesítményű gépekhez azonban bonyolultabb rendszer készül, amelynek betöltése lemezről történik. Ekkor a felhasználó választhat, hogy az adott géphez kapható operációs rendszerek közül melyiket kívánja üzemeltetni. A PC DOS futtatására képes

számítógépeken is általában mód van egyéb operációs rendszerek használatára (CP/M-86, VENIX, XENIX, QNX stb.).

A PC DOS operációs rendszert a Microsoft cég készítette az IBM PC számítógéphez, az IBM megrendelésére. A Microsoft cég megtartotta az operációs rendszer forgalmazói jogát, s a két cég egymástól függetlenül is végzett rajta fejlesztést. Ezért a Microsoft által önállóan forgalmazott MS DOS - s a klónokkal általában ezt a rendszert adják - bizonyos eltéréseket mutat az IBM PC DOS-tól. Leírásukkor az eltéréseket feltüntetjük.

A DOS szó a **Disk Operating System** (lemezes operációs rendszer) rövidítése. Az IBM gépei (PC, PC XT, PC AT, PPC, PS2 stb.), valamint más gyártók IBM kompatibilisnek nevezett termékei hatalmas sikert arattak a professzionális mikroszámítógépek piacán. Bár az IBM más operációs rendszereket is forgalmaz gépeihez, mégis a PC DOS csaknem egyeduralkodóvá vált ezek között. A siker néhány oka: alacsony ár, könnyű kezelhetőség, széleskörű szolgáltatások, az IBM messzemenő támogatása.

A PC DOS nem-multiprogramozható egyfelhasználós operációs rendszer. Mint a neve is mutatja, a billentyűzeten, a képernyőn és a nyomtatón kívül a lemezek - hajlékony- és fixlemezek (Winchesterek) - kezelésére is képes, sőt a rendszer más perifériák vezérlő rutinjaival is kibővíthető. Mi több, legalább egy lemezegységre szüksége van. A parancsokat a billentyűzetről és az ún. parancs-file-okból is elfogadja (interaktív, ill. kötegelt, batch üzemmód).

Lemez- és file-kezelésének főbb jellemzői:

- hierarchikus, fastruktúrájú tartalomjegyzék (directory),
- szekvenciális és közvetlen (direkt) hozzáférést támogató rutinok,
- dinamikus helykezelés a lemezeken (a file-ok mérete változhat),
- file-ok keresésekor globális file-nevek használata,
- egységhibák jelzése, kezelése,
- a programok standard inputjának (billentyűzet), ill. outputjának (képernyő) átirányítási lehetősége (pl.lemezről, ill. lemezre),
- két program összekapcsolása úgy, hogy az egyik program standard outputja a másik program standard inputja lesz (piping)

A PC DOS számos hasznos segédprogramot tartalmaz. A fontosabb segédprogramok funkciói:

- file-ok másolása, összehasonlítása, átnevezése, megjelenítése, nyomtatása,
- floppy lemezek formázása, másolása, összehasonlítása,
- fix lemezek archiválása floppy-ra és visszatöltése,
- szöveges file-ok létrehozása és átszerkesztése,
- gépi kódú programok nyomkövetése, hibakeresés,
- tárgy-(object) programok összeszerkesztése,
- file-ok válogatása, rendezése.

A PC DOS-hoz általában mellékelik a BASIC nyelv interpreterének egy vagy két változatát is. A BASIC programozással e könyvben nem foglalkozunk.

A PC DOS sok tulajdonságát a hagyományos CP/M rendszertől örökölte, de a UNIX operációs rendszer egyes lehetőségeit (hiearchikus tartalomjegyzék, input és output átirányitása, összekapcsolása, vagy a MORE parancs, bizonyos DOS funkciók) *is megvalósítja.*

Túl a felsoroltakon, a PC DOS legfontosabb jellemzője: népszerűsége és elterjedtsége. Ennek köszönhető ugyanis, hogy a legtöbb mikrogépgyártó PC DOS futtatására képes gépeket is gyárt, nagyon sok software-cég pedig mikrogépes programjait először a PC DOS-hoz készíti el. Ha tehát a felhasználó a PC DOS-t választja, biztos lehet benne, hogy a "fő áramlatban úszik", és még hosszú évekig a legszélesebb választékból szerezheti be a feladatai ellátásához szükséges hardware- és software-eszközöket. A saját fejlesztésű programokat is érdemes PC DOS alatt megírni, mert - ha a jövőben más, nagyobb teljesítményű gépre kell áttérnünk - akkor is biztosan választhatunk olyan számítógépet, amelyiken a PC DOS valamelyik változata működik, és így nem vész kárba a gépváltáskor a korábban befektett munka.

A PC DOS az IBM személyi számítógépével, az IBM PC-vel együtt fejlődött. Az eredeti gép még kazettás egység kezelésére is alkalmas volt. Jelenleg hazánkban legelterjedtebb a PC vagy MS DOS 3.10-es változata (1987.május 6-án!), a hazai forgalmazók is ezt szállítják. Azok azonban, akik megtehetik, már térnek át a jelenleg legfejlettebb 3.30-as változatra.

1.2 A PC DOS futtatásának hardware feltételei

A PC DOS futtatásához IBM PC/XT/AT vagy azzal kompatibilis mikroszámítógépre van szükség. Ezek Intel 8086-os típusú, vagy azzal kompatibilis utasításkészletű mikroprocesszorral rendelkeznek.(Néhány kompatibilis típus: 8088, 80286, 80386.) E processzorok jellemzői: tizenhat bites belső szóhossz, nyolc- tizenhat- vagy harminckét bites adatbusz, húsz-huszonnégy bites címbusz, négy-húsz Megahertz-es órajel. A fő processzor mellett a 8087-es (vagy 80287-es) aritmetikai segédprocesszor is alkalmazható, amely jelentősen felgyorsítja a lebegőpontos műveleteket.

Egyes géptípusokon <TURBO> feliratú gomb található, amely megnyomása az órajel sebességét 20-40%-kal megnöveli. Ilyen gépeket általában célszerű mindig „turbo" üzemmódban működtetni.

A számítógép operatív tára csak olvasható (ún.ROM), és írható-olvasható (ún.RAM) memóriából áll. A ROM tár mérete 64 Kbyte alatt van, és a következő funkciókat ellátó rutinokat tartalmazza:

- a számítógép bekapcsoláskori ellenőrzése (ön-teszt),
- input-output alapműveletek végrehajtása a gépbe beépített (beépíthető) perifériákra,
- a számítógép aktuális konfigurációjának felismerése,
- a képernyőn ábrázolható karakterek pontképe (grafikus módban),
- BASIC interpreter (nem minden géptípusnál).

Megjegyezzük, hogy az IBM legújabb, Personal System/2 sorozatában a ROM mérete 128 Kbyte.

A ROM-ba beégetett rutinokat a DOS is gyakran felhasználja, elsősorban a BIOS-nak nevezett rutingyűjteményt, amely az input-output műveleteket végzi. (A BIOS szó a Basic Input Output System input-output alaprendszer rövidítése.)

Az írható-olvasható (RAM) memória mérete általában 256 és 640 Kbyte között változik. *Utólag azonban a RAM tovább bővíthető; az elérhető maximum 640 Kbyte.* Az AT gépeknél további tárbővítés is lehetséges (3 Mbyte-ig), de a DOS az 1 Mbyte feletti tárat nem tudja kihasználni, legfeljebb "virtuális lemezként" (floppy lemez szimulációjára).

A PC DOS-t futtató gépek kötelező perifériái: billentyűzet, képernyő és legalább egy lemezmeghajtó egység. A billentyűzetek különböző betűkészlettel és elrendezésben készülnek, az egyes nemzeti nyelvek sajátosságai szerint. A képernyőt (monitort) a hozzátartozó vezérlő kártyával együtt az alábbi kategóriákba lehet sorolni:

- monokróm vagy színes,
- alfanumerikus vagy grafikus.

A monokróm monitorok csak kétféle színt biztosítanak (például zöldet és fehéret). Az alfanumerikus monitorok csak a gép jelkészletét képező 256 karakter ábrázolására képesek. A grafikus monitorok pontonként is címezhetők, bármelyik pontjuk önállóan is kezelhető (például egy 640-szer 200-as pontmátrixban). Szöveges megjelenítés esetén a legtöbb monitor 25 sorban 80-80 karaktert ír ki.

A következő főbb monitorok és csatolókártyák léteznek:

- monokróm (monokróm, alfanumerikus)
- Hercules monografikus (monokróm, grafikus, 720x350 képpont)
- színes grafikus (640x200 képpont, ff. ált. 320x200 képpont 4 színben)
- nagyfelbontású színes grafikus (640x350 képpont 16 színben)

A legismertebb csatoló kártyák:

MCA (monocrome adapter)
HGA (Hercules graphics adapter)
CGA (color graphics adapter)
EGA (enchanced graphics adapter)
PGA (professional graphics adapter)

Természetesen léteznek más monitortípusok is (pl. az IBM új PS/2 gépcsaládjánál, illetve tervezői - CAD - munkához). A legjobbak felbontása 2048x2048 képpont is lehet 4096 színben.

A lemezmeghajtó egységeket két nagy csoportra osztjuk: **a hajlékony (floppy) és a fix (Winchester) lemezeket kezelő egységekre.** A floppy lemezek cserélhetők,egy- vagy kétoldalasak, tárolási kapacitásuk általában 160-tól 1440 Kbyte-ig terjed. A fix lemezek a meghajtó egységükkel együtt be vannak építve a számítógépbe, nem cserélhetők. Elérési sebességük igen nagy, tárolási kapacitásuk kb. 10-200 Mbyte.

A nem kötelező perifériák közül kiemeljük a **nyomtatókat** és az **RS-232 adaptert** (utóbbi a számítógépek egymással való összeköttetésére, illetve bizonyos perifériák csatlakoztatására szolgál). Elképzelhető - többek között - rajzgép (plotter), digitalizáló rajztábla (digitizing tablet), botkormány (joystick), potenciométer (paddle), egér (mouse), fényceruza (light pen), modem és különböző terminál-emulátor kártyák bekötése is a számítógépbe.

A nagykapacitású Winchesterek fokozatos terjedése növelte az igényt, hogy az azon levő adattartalom menthető legyen **cserélhető mediára** is. Jelenleg nagyon sokféle ún. **streamer-egység** vásárolható az IBM kompatibilis számítógépekhez. Egy-egy kazettára 10, 20, 40 illetve 60 Mbyte-nyi információ menthető ki. Ezekhez az egységekhez külön program tartozik, amelyet az egységhez adnak.

A PC DOS alatt üzemelő számítógépek hardware és software szempontból is igen rugalmas struktúrát alkotnak, ami megkönnyíti a gép hozzáigazítását a változó, növekvő feladatokhoz. **Bővíthető a RAM memória, és szabad kártyapozíciók vannak a gépben, amelyek különböző perifériák illesztésére is felhasználhatók.** A gépet más számítógépekkel össze lehet kapcsolni, és nagy- vagy minigépek intelligens termináljaként üzemeltetni.

A PC DOS is nyílt rendszert alkot, módosítására és bővítésére számos lehetőség kínálkozik (gondoljunk például az új perifériákat vezérlő rutinok beillesztésére).

Végül egy figyelmeztetés az "IBM kompatibilisként" hirdetett számítógépek vásárlói számára: mielőtt ilyen gépet vennénk, győződjünk meg arról (lehetőleg kipróbálás útján), hogy az adott konfiguráción valóban lefutnak-e az alkalmazni kívánt programok (beleértve a PC DOS-t is). Ha ez a feltétel teljesül, még akkor sem lehetünk benne biztosak,hogy más programok - amiket a későbbiekben vásárolnánk - szintén hibátlanul működnek a kiválasztott gépen. Ezért lehetőleg olyan számítógépet vegyünk, amelynek kompatibilitásáról (hasonlóságáról) komoly garancia vagy tapasztalat áll rendelkezésre. Akinek lehetősége van rá, próbáljon írásos szakvéleményt beszerezni a vásárolni kívánt konfiguráció kompatibilitási szintjéről. (DOS, BIOS vagy hardware kompatibilitás)

Ez a figyelmeztetés különösen érvényes a mikrogépes hálózatokra. Sajnos a hazai gyakorlat ezen a téren különösen rossz: házilag barkácsolt kártyák és licensz nélküli szoftverek tömege lepte el az országot. A hálózati hibák nehezen kiküszöbölhetők, s általában véletlenszerűen jelennek meg. A 100%-os kompatibilitás hálózati üzem esetén nélkülözhetetlen!

1.3 A PC DOS verziói

A PC DOS nincs fixen beépítve a számítógépekbe, betöltése lemezről történik, ezért a rendszer továbbfejlesztését, különböző géptípusokhoz való igazítását semmi nem gátolja. Az első IBM PC számítógépekhez eleinte a DOS 1.00, később az 1.10 verzióját árusították. Ezek a változatok még nem kezelték a fix (Winchester) lemezeket. Az IBM PC XT gépek már tartalmaznak fix lemezt, ehhez a géptípushoz dolgozta ki a Microsoft a 2.00 verziót. A DOS 2.00 verziója a fix lemez kezelésén kívül más újdonságot is tartalmaz, például a hierarchikus (fastruktúrájú) directory-t (tartalomjegyzéket), a standard input és output átirányítását vagy összekapcsolását, a háttérben történő nyomtatást, javított

BASIC interpretert stb. A DOS 2.11 verziója a hordozható IBM PC-n (PPC) is futtatható, egyébként felhasználói szempontból teljesen megegyezik a 2.00 verzióval.

Az IBM PC AT nagyobb kapacitású floppy és fix lemezeit csak a DOS 3.00 és későbbi verziói tudják kezelni. A 3.00 verzió a 2.00-hoz képest a következő lényeges továbbfejlesztéseket tartalmazza: virtuális lemezkezelés a RAM-ban, javított programszerkesztő (LINK) és BASIC interpreter, file-ok törlés elleni védelme, file-ok osztott használata (file sharing) stb. A 3.10 verzió pedig a számítógép hálózatba kötésekor futtatott hálózati software-t kiszolgáló rutinokat is tartalmazza, de egyébként megegyezik a 3.00-val. A DOS 3.20-as változata további új utasításokat tartalmaz. A DRIVER.SYS driver lehetővé teszi, hogy különböző típusú hajlékonylemezes egységeket tartalmazzon a rendszer. A régi utasítasítások közül hetet továbbfejlesztettek. A DISKCOMP és a DISKCOPY parancs a 720 Kbyte-os egységre is alkalmazható.

A FORMAT nem formázza meg az aktuális meghajtót, csak ha a lemezegység betűjelét beírjuk a parancsba. Ez bizonyos védelmet jelent a fix lemezegység újraformázása ellen.

Az IBM és Microsoft legújabb közös fejlesztése, a DOS 3.30-as verziója leglényegesebb újítása, hogy lehetővé teszi a 32 Mbyte-nál nagyobb fix lemezek (winchesterek) használatát is.

A felsorolt DOS verziók egymással felülről kompatibilisek, tehát a kisebb számú verzió alatt készült felhasználói programok a nagyobb verziószámú rendszerek alatt is futtathatók, így semmi nem gátolja a fejlettebb verziókra való áttérést.

Az IBM kompatibilis számítógépekhez a legkülönbözőbb DOS változatokat forgalmazzák, amelyek többé-kevésbé eltérnek a PC DOS említett verzióitól. Az IBM PC-khez legjobban hasonlító gépeken azonban a 2.11 vagy 3.00 verziók is lefutnak, ilyenkor érdemes ezeket alkalmazni.

Könyvünk írásakor el kellett döntenünk, hogy melyik DOS változattal foglalkozzunk. Választásunk az IBM által forgalmazott legfeljebb DOS 3.20-as verziójára esett. Könyvünk írásakor sajnos még nem állt rendelkezésre a 3.30-as verzió részletes leírása. A részletes - technikai - kézikönyv akkor még külföldön sem jelent meg. A második szerkesztés során már a 3.30-as újdonságokat is be tudtuk építeni a könyvbe.

Természetesen minden esetben feltüntetjük, hogy a korábbi verziók milyen formában használják a szóbanforgó parancsot. Könyvünk "visszafelé" az IBM PC DOS 2.10 változatig tartalmaz információt (beleértve az MS DOS 2.11, DOS 3.00, DOS 3.10 és DOS 3.20-as változatokat.)

Könyvünk kijelentései és példái tehát csak a DOS 3.30 alatt üzemelő gépekre érvényesek teljes mértékben. Sok IBM kompatibilis gép (pl. Commodore PC-10, PC-20, valamint a magyar PROPER-16 számítógép felhasználóinak kedvéért azonban kitérünk e gépek operációs rendszereinél (MS DOS 2.11, ill. a PROPER-16-nál a PROPOS 3.0) tapasztalható lényeges eltérésekre. Más DOS változatok futtatásakor további különbségekre kell számítanunk, melyek felderítéséhez az adott rendszer referencia-kézikönyvét is elő kell vennünk.

2. PC DOS alapismeretek

Ebben a fejezetben olyan alapkérdésekkel foglalkozunk, amelyek ismerete már az első DOS parancs kiadása előtt nélkülözhetetlen. Ezek a következők:

- a DOS funkcionális részei,
- a DOS lemezkezelése,
- a DOS indítása és újraindítása
- a billentyűzet kezelése DOS-ban,
- a parancsok leírásánál alkalmazott jelölések.

Az első két témát itt még nem tárgyaljuk teljes részletességgel, csak a továbbhaladáshoz szükséges mértékben. A következő fejezettől kezdve pedig már a DOS konkrét alkalmazásával, az egyes DOS parancsokkal foglalkozunk.

2.1 A PC DOS funkcionális részei

A DOS operációs rendszert tartalmazó floppy lemezeken számos program található. Két - a DOS szempontjaból fontos - programot azonban nem láthatunk a tartalomjegyzékben, mert ezek rejtett (hidden) jelzéssel vannak ellátva. A rendszert az operatív tárba betöltő kis program pedig nem önálló file-ként helyezkedik el a lemezen, ezért file-neve sincsen. A DOS programjait funkciójuk szerint a következő csoportokba oszthatjuk:

- betöltő program (betölti a DOS-t a tárba),
- IBMBIO.COM program (a gép ROM tárába beégetett BIOS rutinokkal tartja a kapcsolatot, MS DOS neve: IO.SYS)
- IBMDOS.COM program (a felhasználói programokkal tartja a kapcsolatot, MS DOS neve: MSDOS.SYS).
- COMMAND.COM program (feldolgozza a DOS parancsokat),
- a DOS "külső" parancsait tartalmazó segédprogramok,
- a BASIC interpreter egy vagy két változata, mintaprogramokkal.

A lemezes input-output műveleteket számítógépünk a DOS közvetítésével hajtja végre.

Maga a DOS pedig a következőképpen kerül be a lemezről az operatív memóriába: a gép bekapcsolásakor, valamint a rendszer újraindításakor a ROM tárba fixen beépített program (amely független az operációs rendszerektől) beolvassa a lemez legelső szektorának tartalmát a RAM-ba, és végrehajtja azt. Az első szektorban éppen a betöltő program (boot record) helyezkedik el, amely – DOS lemezek esetén – betölti a tárba az IBMBIO és IBMDOS rendszerprogramokat.

Az IBMBIO.COM program "rejtett" jelzéssel van ellátva, a tartalomjegyzéken nem jelenik meg. (A PROPOS 16 operációs rendszerben: PROBIO.COM) Feladata a ROM-ban elhelyezkedő BIOS (input-output alapműveleteket végző) rutinokkal való kapcsolattartás. Ahogyan maga a BIOS, az IBMBIO program is erősen gépfüggő.

Az IBMDOS.COM program is "rejtett" jelzésű. (A PROPOS l6 operációs rendszerben: PRODOS.COM) Ez a program tartja a kapcsolatot a felhasználói programokkal,

végrehajtja azok kívánságait, közvetít az IBMBIO felé. Az IBMBIO és az IBMDOS betöltésük után végig a gép memóriájában maradnak, csökkentve a felhasználói programok számára fennmaradó szabad területet.

A COMMAND.COM programot parancs-processzornak, parancs-feldolgozónak is nevezik. Ez olvassa be, értelmezi és hajtja végre a billentyűzetről (vagy a parancs-file-okból) az általunk kiadott DOS parancsokat. Az ún. belső parancsok kódját, programját is a COMMAND.COM tartalmazza, a külső parancsokat pedig a lemezen elhelyezkedő DOS segédprogramok futtatásával hajtja végre. A parancsprocesszor kisebb része állandóan a memóriában foglal helyet, de nagyobbik részét szükség esetén a felhasználói programok kitörölhetik, és tárterületét saját céljukra felhasználhatják. Utóbbi esetben a program lefutása után a COMMAND.COM-ot ismét be kell tölteni a rendszerlemezről (ezt a DOS, pontosabban a COMMAND.COM rezidens része automatikusan meg is teszi).

A gyári DOS lemezen található .COM és .EXE kiterjesztésű programokat nevezzük DOS segédprogramoknak, ill. külső parancsoknak. Említettük, hogy a legfontosabb funkciók (belső parancsok) kódját a parancsprocesszor tartalmazza, ezek végrehajtásához nem kell a lemezről programokat beolvasni. A ritkábban igényelt feladatok (külső parancsok) programját azonban nem lett volna érdemes a COMMAND.COM-ba beépíteni, nehogy annak mérete túl nagyra nőjön, és elvegye az operatív tárban a felhasználói programok elől a helyet. Ezért, amikor olyan DOS parancsot adunk ki, amelynek kódja nem szerepel a parancsprocesszorban, a DOS a lemezen keresi meg a parancs nevével megegyező, .COM vagy .EXE kiterjesztésű programot, és elindítja azt.

Ebből is látszik, hogy nincs lényeges különbség a DOS külső parancsai és az egyéb végrehajtható (gépi kódú) programok között; mindkét típusba tartozó programok a lemezen önálló file-ként helyezkednek el (.COM vagy .EXE kiterjesztéssel), és keresésük, lefuttatásuk azonos módon történik.

A DOS parancsok tárgyalásakor mindig megemlítjük majd, hogy belső vagy külső parancsról van-e szó.

Az említett DOS programokon kívül a rendszerlemezeken általában megtaláljuk a BASIC interpreter egy vagy két változatát is.

2.2 A PC DOS lemez-kezelése

A DOS verziói hajlékony (floppy) és fix (Winchester) lemezeket is tudnak kezelni. A DOS-t futtató számítógépek általában egy vagy két floppy-meghajtóval vannak felszerelve (5 1/4 vagy 3 1/2 inch átmérőjű lemezek számára). A fix lemezek használata nem kötelező, de nagyobb adattárolási igények esetén egy vagy két Winchester-meghajtóval is célszerű kiegészíteni a konfigurációt.

A floppy-meghajtók egy vagy két fejjel rendelkeznek, ettől függően a lemezeket csak az egyik vagy mindkét oldalukon tudják írni, illetve olvasni. Az adatok tárolása a floppy lemezeken koncentrikus körökben, ún. sávokban (track) történik. A sávok tovább oszlanak szektorokra. Egy-egy szektorban (a DOS alatt) 512 adat-byte tárolható. A sávok és szektorok száma meghajtó típusonként más és más lehet. A DOS a lemezeket negyven,

egyes esetekben nyolcvan sávosra formázza, sávonként (választhatóan) nyolc vagy kilenc szektorral. A lemezek adattárolási kapacitását így a következő táblázat szemlélteti:

egyoldalas lemez, 40 sáv 8 szektor/sáv: 160 Kbyte
egyoldalas lemez, 40 sáv 9 szektor/sáv: 180 Kbyte
kétoldalas lemez, 40 sáv 8 szektor/sáv: 320 Kbyte
kétoldalas lemez, 40 sáv 9 szektor/sáv: 360 Kbyte
kétoldalas lemez, 80 sáv 9 szektor/sáv: 720 Kbyte,
kétoldalas,nagykapacitású lemez, 80 sáv 15 szektor/sáv: 1.2 Mbyte

Az egyfejes meghajtókat az újabb modellekben már nem alkalmazzák, a nyolcszektoros tárolás pedig csak a DOS korábbi verzióival fennmaradó kompatibilitás érdekében maradt fenn. A leggyakoribb kapacitás így a 360 Kbyte, illetve az AT gépeken az 1.2 Mbyte. A 720 Kbyte-os lemezegység támogatása elsősorban az egyéb rendszerekkel való adatcsere biztosítására szolgál.

(Fontos tudni, hogy az IBM a 80 sávos és 9 szektor/sáv típusú lemezegységeket csak a 3.10-es verziótól támogatja, elsősorban a külső egységek kezelésére. A PROPER gépek 80 sávos lemezformázási módja nem kompatibilis a DOS-éval, így ezeket a PC-DOS alatt nem lehet olvasni, illetve írni.)

Az IBM legújabb, PERSONAL SYSTEM/2 számítógépcsaládja már nem 5 1/4, hanem 3 1/2 inch-es floppy meghajtóegységeket tartalmaz. Ezek kétoldalas, 80 sávos, tehát 720 Kbyte tárolókapacitású lemezeket használnak. A PS/2 gépcsaládhoz külső egységként csatlakoztathatók 5 1/4 inch-es egységek.

Ha valaki 80 sávos egységét tartalmazó PROPER gépét át akarja állítani a PROPOS operációs rendszerről PC-DOS-ra, a következőt kell tennie:

1. Az eddigi 80 sávos PROPOS formájú lemezeinek tartalmát át kell másolnia 40 sávos (PC-DOS kompatibilis) lemezekre.
2. A fix lemezt el kell mentenie 40 sávos lemezekre.
3. A DOS 3.20 rendszerlemezről újraindítva (boot-olva) a rendszert, újra kell formáznia a fix lemezt.
4. A fix lemezre vissza kell tölteni a file-okat a mentési lemezről.

A DOS 3.00 verziója már az 1.2 Mbyte-os (1200 Kbyte-os) floppy-k kezelésére is képes, de más kapacitású meghajtók is forgalomban vannak.

Az egyes meghajtókon létrehozott lemezek nem mindig olvashatók be más típusú meghajtókon. *Az 1.2 Mbyte-os meghajtón például be lehet olvasni 360 Kbyte-os lemezeket, de fordítva ez már nem igaz.* Ügyelni kell arra is, hogy oldalszámban és tárolási sűrűségben megfelelő lemezeket vásároljunk az adott floppy-meghajtóhoz. **Speciálisan AT-ra készült lemezeket - azok mágneses tulajdonságai miatt - az XT-s gépeken nem lehet használni.**

Sajnos az 1.2 Mbyte-os egységek fizikailag eltérő felírási módja miatt hiába írunk fel XT-n megformázott 360 Kbyte-os lemezekre az AT-n, azt nem mindig tudjuk utána az XT-n elolvasni. Emiatt az XT-AT kapcsolat - kivéve a nagyon ritkán található igazán jól beállított AT-kat -többnyire egyoldalú: az AT tudja olvasni az XT-n felírt lemezeket, fordítva ritkaságnak számít.

A floppy lemezek oldalán egy kisebb bevágást találunk, erre ragasztható a lemezekhez mellékelt alumínium vagy papír írásvédő csík. A csík felragasztásával lehet a lemezt megvédeni törlés és felülírás ellen.

Ilyenkor a gép csak olvasni tudja az adott lemezt. A floppy-lemezek sajnos sérülékenyek, érzékenyek a hőre, porra, mágneses térre, kézzel történő érintésre. Ezért a teljes biztonság érdekében rendszeresen készítsünk másolatot a fontos programokról, adatokról.

A 3 1/2 inch-es floppy lemezek legfőbb erénye, hogy kevésbé sérülékenyek, mivel az egységből kivéve mágneses részük teljesen fedett. (Az író-olvasó rést egy csúszófedő védi, a lemezt a meghajtóba téve teszi csak szabaddá a rést.)

Az üres floppy-kat első felhasználásuk előtt a DOS megfelelő parancsával meg kell formáznunk. E művelet során alakítja ki a DOS a lemezen a sávokat és szektorokat (egyelőre üres adattartalommal). A formázáskor megadhatunk bizonyos opciókat, amelyek a létrehozandó sávok és szektorok számát befolyásolják. Előírhatjuk továbbá, hogy az új lemezre átmásolódjon-e maga a DOS rendszer.

A DOS betöltő programja (boot record) a formázáskor minden lemezre átkerül, de az IBMBIO és az IBMDOS programok csak abban az esetben, ha a formázó parancsban a DOS rendszerprogramok másolását is előírtuk. (A DOS külső parancsait tartalmazó segédprogramokat a formázás után kiadott másoló paranccsal lehet a lemezre vinni.) A DOS-t természetesen csak az említett három alapprogramot tartalmazó lemezekről lehet betölteni, elindítani. Ha a betöltő program nem találja az alapprogramokat, akkor hibajelzést ad. A rendszer elindítása után már nem kell, hogy fent legyen a DOS-t tartalmazó lemez.

A fix (Winchester) lemezek közös tengelyre szerelt merev korongokból állnak, melyek egybe vannak építve a meghajtóegységgel, tehát nem cserélhetők. Az egyes felületek - floppy-khoz hasonlóan - sávokra és szektorokra oszlanak. A szektorok itt is 512 adat-byte tárolására képesek. Minden felülethez külön író-olvasó fej tartozik. Az egymás fölött elhelyezkedő (fejmozgás nélkül elérhető) sávokat együttesen cilindereknek hívjuk. A lemezek szokásos kapacitása 10, 20, 40 Mbyte, de ennél nagyobb méretek is előfordulnak.

Felhívjuk a figyelmet arra a tényre, hogy a winchestereknek kétféle kapacitásuk létezik, egy formázatlan (nyers) és egy formázott. A leggyakrabban használt winchester típus 611 cilindert tartalmaz. Ennek formázatlan kapacitása 27 Mbyte, formázott kapacitása azonban csak valamivel több, mint 20 Mbyte. Ha tehát egy szállító 27 Mbyte-os winchester-t ígér, akkor a DOS alatt - formázva - 20 Mbyte tárolókapacitásunk lesz.

A fix lemezeket az első felhasználásuk előtt partícionálni és formázni kell (a DOS rendszerprogramok átmásolásával vagy anélkül).

A partícionálás során oszthatjuk fel a lemez területét a különböző operációs rendszerek között. Ha csak a DOS-t használjuk, egyetlen partíciót képzünk. Célszerű a számítógépünk első fix lemezére átvinni a DOS-t, mert ebben az esetben a rendszer betöltése floppy lemez behelyezése nélkül, a fix lemezről történhet.

A programok és az adatok a hajlékony és a fix lemezeken is file-okban helyezkednek el. Minden file-nak egyedi nevet kell adnunk; egy lemezen (pontosabban egy lemez adott fő- vagy alkönyvtárában) nem lehet két azonos nevű file.

A lemezeken található szektorok többségét a DOS a file-ok tárolására használja fel. A DOS alapprogramjai is (az egy szektoros betöltő rekordon kívül) file-okat képeznek, bár speciális jellegük következtében nem mindegyik jelenik meg a lemez tartalomjegyzékének kiíratásakor.

Honnan tudja a DOS, hogy egy lemezen milyen nevű file-ok találhatók, és ezek melyik lemezfelületen (melyik író-olvasó fej alatt), melyik sávokon, mely szektorokban helyezkednek el? Ezeket az adatokat a rendszer a lemez elkülönített részén kialakított tartalomjegyzékben (directory-ban) tartja nyilván. A tartalomjegyzék bejegyzései gyakran változhatnak, hiszen a file-okat módosíthatjuk, törölhetjük, átnevezhetjük, és új file-okat hozhatunk létre. A DOS mindig naprakész állapotban tartja a tartalomjegyzéket, így a felhasználónak nem kell az adatok fizikai elhelyezkedésével (fej-, sáv- és szektor-információkkal) foglalkoznia.

A nagykapacitású fixlemezek alkalmazásakor nem lenne célszerű az esetleg többszáz file-t ömlesztve egyetlen tartalomjegyzékbe bejegyezve tárolni (igen nehéz volna így a keresés, az eligazodás). Ezt a problémát küszöböli ki a hierarchikus (fastruktúrájú) tartalomjegyzékek használatának lehetősége. Ezzel a módszerrel elkülönítve kezelhetjük az egyes felhasználókhoz (vagy az egyes programokhoz, rendszerekhez) tartozó file-okat.

A tartalomjegyzéken kívül még egy speciális táblázatra van szüksége a DOS-nak, amely a lemez szektorainak (illetve a több szektorból álló, cluster-nek nevezett egységeknek) foglaltságáról tartalmaz információkat. Ez a file-helyfoglalási táblázat (angol neve: FAT, File Allocation Table). A FAT a lemeznek erre a célra fenntartott szektoraiban helyezkedik el, és a benne szereplő adatok alapján foglalja le (ill. szabadítja fel) a DOS a file-ok számára szükséges (ill.már nem igényelt) lemez-területeket.

2.3. A PC DOS indítása

A számítógépünk bekapcsolása után el kell indítanunk a DOS-t, mielőtt az operációs rendszert igénylő programokat futtatnánk. A rendszer betöltése hajlékony vagy fix lemezről történhet, a felhasználóra vár azonban a feladat, hogy szükség esetén a mindennapi használatra szánt rendszerlemezt kialakítsa. A rendszerlemez létrehozásáról és újraindításáról szól ez a rész.

2.3.1. A DOS elindítása

A DOS operációs rendszer elindításához szükségünk van a DOS rendszerlemezre (esetleg másolatára) vagy arra, hogy a számítógépben található első fix lemezegység aktív partícióján ott legyen a DOS. Arról, hogyan érhetjük el, hogy a DOS felkerüljön a fix lemezre, a következő részben szólunk.

Bekapcsolás után a gépbe beépített program először ellenőrzést végez, teszteli a gép fontosabb egységeit. Ha a teszt eredményes volt (nem jelenik meg hibaüzenet), akkor megkezdődik az operációs rendszer betöltése. A gép először az "A" jelű meghajtóval próbálkozik, de, ha a meghajtóban nincs lemez (vagy az egység ajtaja nyitva van), akkor az első (általában "C" jelű) fix lemezről kísérli meg betölteni a rendszert. Ha a hajlékonylemezes

egységbe helyezett lemez nem tartalmaz önindítható programot, akkor a következő üzenet jelenik meg:

Non-System disk or disk error
Replace and strike any key when ready.
(Nem rendszerlemez vagy lemez-hiba.
Cserélje ki a lemezt, majd nyomja le bármelyik billentyűt.)

Ha ez a hiba lép föl, tegyük be a DOS rendszerlemezt az "A" meghajtóba, és nyomjunk le egy billentyűt. (Ha a winchester-en van a DOS, vegyük ki a lemezt az egységből és nyomjunk le egy billentyűt.

A DOS rendszerlemezen kívül egyéb automatikus indítású programok is vannak. Ezek, hasonlóan a DOS-hoz, a gép bekapcsolásával egyidejűleg betöltődnek a memóriába és elindulnak.

Ha az "A" meghajtó üres, és a géphez nem tartozik fix lemez (vagy a fix lemez nem tartalmaz operációs rendszert), de a gép ROM tárába be van égetve a BASIC, akkor a DOS helyett a BASIC interpreter lép működésbe.

Tudnunk kell, hogy a BASIC-nek ez a minimális változata (az ún. kazettás BASIC) nem képes sem a lemezes file-ok kezelésére, sem a kiterjesztett BASIC változat néhány fontosabb (pl. grafikus) funkciójának ellátására.

Az operációs rendszer sikeres betöltése után a DOS először a CONFIG.SYS file létezését ellenőrzi. Ha megtalálja, akkor a benne levő paraméterek alapján konfigurálja a rendszert, illetve betölti és inicializálja a benne definiált periféria-kezelő (device driver) programokat. Ezután vizsgálja meg, hogy az AUTOEXEC.BAT nevű file létezik-e a főtartalomjegyzékben.

Ha igen, akkor sorban végrehajtja az abban szereplő parancsokat. Tipikus esetben az AUTOEXEC futtatja le a rendelkezésre álló billentyűzetet kezelő segédprogramot (ha nem amerikai a billentyűzet), megkérdezi továbbá a dátumot és az időt. (A parancs-file-okról a 7. fejezet tartalmaz részletes leírást.)

Ha az indító lemezen nincs AUTOEXEC.BAT file, akkor a DOS csak a dátumot és az időpontot kérdezi meg a felhasználótól, erre célszerű mindig pontos választ adni. A dátumot a következőképpen kéri be a DOS:

Current date is Tue 1-01-1980
Enter new date:

Az aktuális dátumot hónap, nap, év sorrendben kell beírni (hacsak a dátum formátumát át nem állítottuk). Az évszázad, valamint a felesleges nullák elhagyhatók. 1986 június 12-én például így kell válaszolni:

6-12-86 < ENTER >

Az időpontot a DOS így kérdezi meg (a konkrét számok a példától természetesen eltérhetnek):

Current time is 00:00:02.13
Enter new time:

Az aktuális időt óra, perc másodperc, századmásodperc sorrendben lehet beírni, a gyakorlatban a másodpercet és a századmásodpercet elhagyhatjuk. Délután negyed háromkor így kell megadni a pontos időt:

14:15 < ENTER >

(A DATE és a TIME parancsokat részletesen a 6.4 és a 6.5 pontokban ismertetjük.)

Sok IBM PC XT-t és AT-t már belső órával (ami a gép működése közben töltődő akkumulátorral működik) szállítanak. Ebben az esetben az AUTOEXEC.BAT-be célszerű elhelyezni azt a programot, amelyik a belső óra idejét átírja az operációs rendszer órájába.

A dátum és az időpont beírása, illetve az AUTOEXEC file utolsó parancsának végrehajtása után a DOS megáll, és a parancsunkat várja. Ebben az állapotban a képernyő utolsó parancs-sorában csak a DOS prompt-ja (készenléti jele) látható:

A > (hajlékonylemezről) vagy
C > (fix lemezről)

A prompt jelzi, hogy a DOS a parancsainkat várja. A felszólító jel első betűje az ún. aktuális meghajtóra utal (erre a meghajtóra vonatkoznak majd a parancsok, ha nem szerepel bennük más meghajtó jele).

A prompt alakja - az AUTOEXEC.BAT file tartalmától függően - változhat, de a gyárilag szállított rendszerek általában nem tartalmaznak AUTOEXEC.BAT file-t, és így a DOS azonnal bejelentkezik. (Az AUTOEXEC.BAT file tetszőleges parancsokat is tartalmazhat. Ebben az esetben ezek a parancsok is végrehajtódnak, pl. elindulhat egy program, így előfordulhat, hogy a DOS prompt-ja egyáltalán nem jelenik meg.)

A DOS 3.00-ás és magasabb verziói a 10 Mbyte-nál nagyobb fix lemezegységeket a DOS 2.xx verzióitól eltérően kezelik. Ha például az "A" egységben levő DOS 2.11-es operációs rendszert elindítjuk és a "C" fix lemezegységet 3.00 alatt formáztuk meg, akkor üres tartalomjegyzéket kapunk! Ha nem tudjuk, hogy mi van egy lemezen, akkor mindig először onnan kísérelljük meg betölteni az operációs rendszert!

2.3.2 Ismételt indítás (restart)

Az előző pontban leírtak alapján bármikor el tudjuk indítani az operációs rendszert, ha a számítógép kikapcsolt állapotban volt. Előfordulhat azonban, hogy a gépet már bekapcsoltuk, és a DOS-t csak később akarjuk elindítani, vagy újraindítani. Néhány példa erre a szituációra:

- a gépen kazettás BASIC fut, és el akarjuk indítani a DOS-t,
- a jelenleg üzemelő DOS-tól eltérő verziószámú rendszert akarunk futtatni,
- módosítottuk a CONFIG.SYS leírást, és ezt aktivizálni szeretnénk,
- a DOS alatt futó program "lemerevedett", és más módszerrel nem tudunk kilépni belőle, ezért kell újraindítani a DOS-t.

A DOS újraindításához nem szükséges a számítógépet ki- és bekapcsolni. Jobban kíméli a gépet, és rövidebb ideig tart (kiesik a bekapcsoláskori ellenőrzés), ha az alábbi módszert választjuk:

Ha az újraindítást floppy-ról végezzük, helyezzük be a DOS rendszerlemezt az első meghajtóba. Fix lemez esetén erről történő indítás esetén nyissuk ki az első floppy-meghajtó ajtaját. Nyomjuk le ezután egyszerre a következő három billentyűt:

<Alt>, <Ctrl>, <Del>

A DOS újraindítása után megismétlődik a CONFIG.SYS és az AUTOEXEC.BAT file-ok feldolgozása is.

A fenti billentyűzetkombinációt szoftver úton érzékeli a rendszer, ezért ha a megfelelő megszakítási címeket valamelyik program átállította, a megnyomásnak semmilyen hatása sincs.

A gépek egy része tartalmaz <RESET> gombot. Ez nem a billentyűzeten, hanem magán a gépen található. Ennek megnyomása valóban egy „hardver reset" jelet generál. Ebben az esetben a gép teljes inicializálása megismétlődik.

2.3.3. Egyéb tennivalók

Az operációs rendszer eredeti (gyári) lemezeiről célszerű másolatot készíteni és azt használni. Egyes rendszerekhez szállítanak olyan parancs-file-t, ami ezt a feladatot elvégzi, de a legtöbb esetben magunknak kell ezt megtenni az alábbi módon:

1. Indítsuk el a DOS-t az eredeti lemezről (helyezzük a floppy-t az első meghajtóba, majd kapcsoljuk be a gépet)

2. Amikor megjelenik a DOS prompt-ja

A>

készítsünk másolatot a gyári lemezről. (Lehet, hogy a gép először a dátumot és az időpontot kérdezi meg. A kérdésekre most nem kell válaszolni, nyomjuk le az <ENTER> billentyűt.) Ha két meghajtónk van, tegyünk a másodikba egy üres lemezt (lehet formázatlan is). Ha csak egy meghajtóval rendelkezünk, akkor az üres lemezt helyezzük készenlétbe. Írjuk be ezután a következő parancsot:

DISKCOPY A: A:<ENTER>

illetve két meghajtó esetén:

DISKCOPY A: B:<ENTER>

(Az <ENTER> jelölés arra utal, hogy le kell nyomni az <ENTER> billentyűt. Vigyázzunk, mert a billentyűk felirata nem szükségszerűen egyezik meg a képernyőn megjelenő betűkkel, pl. az Y és a Z betűket esetleg fordítva kell használnunk.) Egymeghajtós gépeknél a gyári és az új lemezt esetleg többször is cserélni kell a meghajtóban, erre a DOS fel is szólít. Végül megjelenik az alábbi kérdés:

Copy another (Y/N)?

Nyomjuk le válaszként az <N> billentyűt. (A DISKCOPY parancs részletes leírása a 4.12 pontban található.)

3. Tegyük az eredeti DOS lemezt biztonságos helyre és a további munkát a másolattal folytassuk.

4. Készítsük el a DOS másolaton az önindító (AUTOEXEC.BAT) file-t. Az alábbi példában feltételezzük, hogy német billentyűzet tartozik a géphez. Ha amerikai billentyűzetünk van, egyszerűen hagyjuk ki a KEYBGR szót az eljárásból. Ha a billentyűzet más típusú, akkor a KEYBGR helyett a megfelelő segédprogram nevét kell leírni. Helyezzük az első meghajtóba a DOS másolatot, és írjuk be a következőket:

```
COPY CON AUTOEXEC.BAT <ENTER>
KEYBGR <ENTER>
DATE <ENTER>
TIME <F6> <ENTER>
```

(Az <F6> jelölés arra utal, hogy le kell nyomni az <F6> billentyűt.) Ragasszunk ezután írásvédő csíkot a lemezre, és ezentúl használhatjuk a DOS indítására.

2.3.4 Fix-lemezegység használata

Ha a gépünk fixlemez-meghajtóval is fel van szerelve, akkor célszerű az első (általában "C" betűjelű) fix lemezből rendszerlemezt készíteni, hogy a DOS-t a hajlékonylemezes egység használata nélkül is el lehessen indítani.

A fix lemezeket használatbavételük előtt partícionálni és formázni kell. A partícionálást (a lemez területének felosztását a különböző operációs rendszerek között) az FDISK paranccsal lehet elvégezni. Az alábbi példában a lemez teljes kapacitását a DOS alá rendeljük, mert - egyelőre - csak a DOS operációs rendszert akarjuk használni. Tegyük be az előbbiek során készült DOS másolatot az első floppy-meghajtóba, majd írjuk be:

```
FDISK <ENTER>
```

A képernyőn megjelenő menü első pontját választjuk (DOS partíció létrehozása):

```
1 <ENTER>
```

Az ezután megjelenő újabb menüből ismét az (1.) pontot válasszuk ki. (Elsőrendű DOS partíció létrehozása.) *A 3.30-asnál korábbi verzió esetén ez a kérdés kimarad.*

A következő kérdésre (a teljes lemezt a DOS-szal akarja használni?) igenlő választ adunk:

```
Y <ENTER>
```

A képernyőn a következő üzenet jelenik meg:

System will now restart.
Insert DOS diskette in drive A:
Press any key when ready...

Tetszőleges billentyű megnyomására a rendszer újraindul. Ha az első (C jelű) lemezegységet particionáltuk, akkor a hajlékonylemezes egységbe be kell helyeznünk a DOS rend-

szerlemezt, s csak ekkor nyomjunk meg egy billentyűt. Ezután a program újraindítja a DOS-t, amely - a sikeres partícionálás után - már kezelni tudja a fix lemezt.

(Az FDISK parancs részletes ismertetése az 5.1 pontban olvasható. A fix lemezek további partícionálását is az 5.1 pont alapján kell elvégezni.)

Ezután elkezdhetjük a fix lemez formázását. Az első (példánkban "C" jelű) lemezre átmásoljuk a DOS-t is. A többi fix lemez esetében erre nincs szükség. Írjuk be a következőket:

FORMAT C:/S/V <ENTER>

Fix lemez formázásakor a rendszer figyelmeztető üzenetet küld, majd visszakérdez, hogy biztosan ezt akarjuk? Az "Y" billentyű megnyomására megkezdődik a formázás, amelynek előrehaladását a képernyőn is nyomon tudjuk követni. (Csak 3.20-as vagy későbbi verzió esetén.) Amikor a formázással végezve a gép a fix lemeznek adandó nevet (volume label) kérdezi, írjunk be egy maximum tizenegy betűs nevet (pl. lacilemeze), majd nyomjuk le az <ENTER>-t.

A fix lemez megformázása a lemez (pontosabban a DOS partíció) jellemző adatainak kiírásával ér véget:

21309440 bytes lokal disk space
79872 used by system (függ a használt DOS verziójától!)
21229558 bytes available on disk

A rendszer által elfoglalt hely függ a DOS használt verziójától. Például DOS 3.00 esetén csak 61440 byte.

Végül másoljuk át a DOS-t tartalmazó lemezről a DOS segédprogramjait:

COPY *.* C: <ENTER>

Ezután - a gép legközelebbi bekapcsolásakor - a 2.3.1-ben leírt eljárással már a fix lemezről is indíthatjuk a DOS-t.

A fenti eljárás valamennyi DOS file-t a C: lemezegység főtartalomjegyzékében helyezi el. Ennél célszerűbb a következő parancsok kiadásával egy DOS nevű alkönyvtárat létrehozni, és abban elhelyezni a DOS file-jait:

COPY COMMAND.COM C: <ENTER>
MD C:\DOS <ENTER>
CD C:\DOS <ENTER>
COPY A:*.* C:\DOS <ENTER>

Ebben az esetben azonban az AUTOEXEC.BAT file-ban el kell helyezni a

PATH C:\DOS

parancsot, hogy az operációs rendszer mindig megtalálja a DOS külső parancsait. Az AUTOEXEC.BAT file csak akkor működik rendeltetés-szerűen, ha a főtartalomjegyzékben helyezzük el!

2.4 A billentyűzet kezelése, DOS parancsok kiadása

Mielőtt dolgozni kezdenénk a DOS-szal, meg kell ismerkednünk a billentyűzet használatával, az egyes billentyűk funkciójával. Ebben a pontban csak a DOS billentyűzet-kezelésével foglalkozunk, és felhívjuk arra a figyelmet, hogy más programok (pl. a BASIC interpreter, vagy a különböző szövegszerkesztők) a leírtaktól teljesen eltérő funkciót rendelhetnek az egyes billentyűkhöz. Ezért az alábbiakban ismertetett módszereket csak a DOS parancsok kiadásakor és a DOS segédprogramok, köztük a sorszerkesztő (EDLIN) és a hibakereső program (DEBUG) futtatásakor alkalmazhatjuk változtatás nélkül.

A PC DOS-t futtató számítógépekhez **különböző billentyűzeteket forgalmaznak.** *Nálunk a leggyakoribb az amerikai és a német elrendezésű billentyűzetek használata.* Ezek nemcsak a jelek elhelyezésében térnek el egymástól, hanem a következő szempontokból is különböznek:

- A német billentyűzeten bizonyos speciális jelek (pl. kapcsos zárójelek) helyett ékezetes betűk (ö, ü, ä) találhatók.
- A német billentyűzeten - az amerikaival ellentétben - egyszerűen előállíthatók egyes ékezetes betűk (köztük a magyar á, é, í, ó, ú, és É).
  Ezek képernyős megjelenítéséhez először le kell nyomnunk az ékezet-billentyűt, majd azt a gombot, amelyiken az alapjel szerepel. Így például az ó betűt az ' (aposztrof) és az o billentyűk egymásutáni lenyomásával állíthatjuk elő.
- A német billentyűzeten két speciális jel (a @ és a \) a billentyűk oldalára van írva. Ezeket úgy lehet megjeleníteni, hogy az <ALT> és a <CTRL> billentyűk lenyomva tartása közben megnyomjuk az adott jelet ábrázoló billentyűt.
- Az Y és Z betűk fordítva - a magyar írógép-szabványnak megfelelően - vannak.

A billentyűzet nagyrésze az írógépről ismerős betűket, számokat és jeleket tartalmazza. A szóközbillentyű (space) is a megszokott helyén van. Vigyázni kell azonban, nehogy a felső sorban megtalálható 0 es 1 számokat az o, illetve l betűkkel helyettesítsük.

A nagybetűk, valamint a billentyűkön felülre írt jelek előállításához a <Shift> billentyűt kell nyomva tartani a megfelelő gomb lenyomása alatt. (A <Shift> az írógép nagybetűváltójának felel meg, a billentyűzet bal és jobb oldalán található. Jele: Shift vagy egy felfelé mutató nyíl.) A DOS parancsok kiadásakor egyébként mindegy, hogy nagy- vagy kisbetűket használunk, sőt a DOS még a vegyes írásmódot is elfogadja. Amikor viszont szövegeket, forrásprogramokat állítunk elő pl. az EDLIN programmal, akkor már értelme lehet a kis- és nagybetűk megkülönböztetésének.

Ha tartósan csak nagybetűket akarunk használni, nyomjuk le a <Caps Lock> (nagybetűzár) billentyűt. Ezután - egészen e billentyű újbóli megnyomásáig - valamennyi betű nagybetűs formában jelenik meg. (Ez csak a betűkre vonatkozik; a billentyűkön felülre írt speciális jelek előállításához továbbra is le kell nyomni a <Shift>-et!) Ha a <Caps Lock> lenyomása után kisbetűt akarunk írni, a <Shift> billentyűt kell használnunk.

A betűktől jobbra helyezkedik el az igen fontos <ENTER> billentyű. (Egyes gépeken a RETURN nevet viseli.) Minden DOS parancs beírása után le kell nyomnunk az <ENTER>-t, innen tudja a DOS, hogy vége a parancs beírásának, és a végrehajtást csak ezután kezdi meg.

Az < ENTER > -től jobbra található numerikus billentyűknek (0-tól 9-ig, valamint a tizedespont) kettős szerepe van. Alaphelyzetben nem a számok megjelenítésére szolgálnak, hanem a feliratuk alsó sorában szereplő funkciókat hajtják végre. Átkapcsolásuk számírásra, majd visszakapcsolásuk a < Num Lock > (számzár) billentyűvel történhet. DOS alatt e billentyűket általában az alaphelyzetükben használjuk.

A numerikus billentyűk mellett látható + és - billentyűk csak megismétlik a klaviatúra középső részén is megtalálható azonos jeleket.

Ennyi információ birtokában már bátran kiadhatjuk az első DOS parancsot. Tegyük fel, hogy az AUTOEXEC.BAT nevű file-t akarjuk az egyik floppy-ról a másikra átmásolni. Erre a célra a COPY parancs szolgál, amelyet részletesen a 3. fejezetben ismertetünk. Ezt kellene tehát beírnunk:

COPY A:AUTOEXEC.BAT B: < ENTER >

Ha a fenti parancsot hibátlanul írtuk le, biztosan sikerül a másolás. Mit tehetünk azonban, ha - még az < ENTER > lenyomása előtt - észrevesszük, hogy gépelési hibát vétettünk? A legegyszerűbb javítási mód, ha a < Backspace > (visszalépés, visszatörlés) billentyűvel töröljük a parancs hibás részét, majd beírjuk a helyes szöveget. (**A < Backspace > billentyű a legfelső sorban helyezkedik el, jele általában egy balra mutató nyíl.** Minden lenyomásakor egy-egy jelet töröl visszafelé, azaz balra.) Ha például az előző parancsot véletlenül így írtuk (kihagyva az E betűt):

COPY A:AUTOXE

a javításhoz nyomjuk le kétszer a < Backspace > -t (az utolsó két hibás betű törlésére):

COPY A:AUTO

majd folytassuk a parancsot:

COPY A:AUTOEXEC.BAT B: < ENTER >

(Megjegyezzük, hogy a numerikus billentyűzeten a balra mutató nyilat - és a 4-es számot - ábrázoló billentyű is a < Backspace > -szel azonos szerepet tölt be.)

Ha a hibás parancs után már lenyomtuk az < ENTER > -t, akkor általában maga a DOS veszi észre a hibát (ha értelmetlen utasítást kapott), ilyenkor hibajelzést ad. Ha viszont valahogyan értelmezni tudja a beírt szöveget, akkor elkezdi a végrehajtást.

Az éppen futó programot a következőképpen lehet félbeszakítani (ha tévedésből adtuk ki a parancsot, vagy már nincs szükség a program további futására): tartsuk lenyomva a < Ctrl > billentyűt, majd nyomjuk meg a < Scroll Lock > gombot (oldalán többnyire látható a Break, megszakítás felirat). Ez a legtöbb esetben félbeszakítja az adott program futását. (Ugyanez a hatása a < Ctrl > -C kombinációnak is.)

Tegyük fel, hogy beírtunk egy DOS parancsot, de még nem "küldtük el" az < ENTER > megnyomásával. Rájövünk azonban, hogy a parancs végrehajtására most nincs szükség, vagy látjuk, hogy a parancs annyi hibát tartalmaz, hogy még korrigálni sem érdemes. Hogyan törölhetjük ki az egész sort? Az egyik lehetséges, már ismert megoldás, ha annyiszor lenyomjuk a < Backspace > billentyűt, ahány betűt (karaktert) a parancs tartalmazott. A gyorsabb módszer: az < Esc > feliratú billentyű lenyomása; ez ugyanis az egész sort törli.

A hibás sor ugyan a képernyőn marad (egy \-rel, blackslash-sel lezárva), de végrehajtására nem kerül sor, hanem a gép új sorba lép, és következő parancsunk beírására vár. (Az Esc felirat az angol escape = menekülés, kilépés szó rövidítése.)

Léteznek az eddig leírtaknál bonyolultabb, de hatékonyabb szerkesztési műveletek is, amelyek kihasználják, hogy a DOS "emlékezik" az utoljára beírt parancs-sorra, és így annak szövegét a következő parancs kiadásakor - részben vagy egészben - újra felhasználhatjuk. Ezeket a módszereket a DOS parancsok kiadásánál is felhasználhatjuk, de fontosabb szerepet a sorszerkesztő (EDLIN) program futásánál játszanak. Ezért a funkciós billentyűk, az <Ins> és a <Del> gombok ismertetését a fejezet első olvasásakor bátran átlapozhatjuk. Az EDLIN használata előtt viszont érdemes e billentyűkkel is megismerkedni.

A klaviatúra bal oldalán (egyes billentyűzeteken ettől eltérő helyen) található <F1>, <F2>,...,<F10> feliratú gombokat **funkciós billentyűknek nevezzük** (az egyes programok ugyanis különböző funkciókat rendelhetnek hozzájuk). **A DOS-ban csak az első hat funkciós billentyű használható.**

Az <F1> billentyű lenyomásának hatására az előző DOS parancs (vagy EDLIN sor) **egy-egy betűje** (karaktere) **ismétlődik meg.** Ha pl. két file-másolást végzünk egymás után, és a DOS a

COPY A:AUTOEXEC.BAT B:

parancsot már végre is hajtotta, akkor a következő másoló parancs beírását az <F1> billentyű megkönnyíti.

Tegyük föl, hogy most a COMMAND.COM nevű file-t akarjuk átmásolni az A meghajtóról a B-re. Az előző parancsból így változtatás nélkül átvehetjük az első hét betűt (COPY A:), ehhez nyomjuk le hétszer az <F1> gombot. A képernyőn ez jelenik meg:

COPY A:

A parancs többi részét a szokásos módon kell beírnunk:

COPY A:COMMAND.COM B:<ENTER>

(Megjegyezzük, hogy a numerikus billentyűzeten a jobbra mutató nyilat és a 6-os számot ábrázoló billentyű is az <F1>-gyel azonos szerepet tölt be.)

A fenti feladatra az <F2> billentyűt is felhasználhattuk volna. Az <F2> ugyanis szintén az előző DOS parancs karaktereit ismétli meg, de nem egyesével, hanem az általunk megadott karakterig. **Az <F2> leütése után mindig meg kell nyomni annak a karakternek a gombját is, amelyiknek az első előfordulásáig a másolást kérjük** (ez a karakter már nem vesz részt a másolásban).

A COMMAND.COM másolására vonatkozó parancsot az <F2>-vel a következőképpen adhatnánk meg (feltéve, ha az előző parancs az AUTOEXEC.BAT file másolása volt). Az <F2> lenyomása után meg kell adnunk azt a karaktert, ahol a másolást be kívánjuk fejeztetni. A megadott karakter már nem vesz részt a másolásban, ezért az AUTOEXEC szó első A betűjét jelölhetnénk ki (ez az első betű, amelyre már nincs szükségünk). A másolás viszont mindig a kijelölt karakter első előfordulásáig tart, ezért az A betű lenyomása

esetén csak a COPY karaktersorozat másolódna át. Érdemesebb ezért az <F2> után a ":" (kettőspont) billentyűt megnyomni, mert ekkor ez jelenik meg a képernyőn:

COPY A

A parancs további részét a szokásos módon be kell írnunk:

COPY A:COMMAND.COM B:<ENTER>

Az <F3> billentyű teljes egészében megismétli az előző DOS parancsot, feltéve, hogy a billentyű lenyomásakor a sor első pozíciójánál tartunk. Ha a parancs-sorban már tovább léptünk, akkor csak az adott pozíciótól kezdve másolja át az előző parancs karaktereit. Az <F3> tehát igen kényelmes eszköz a parancsok változatlan megismétlésére is.

Az <F4> billentyű az <F2> ellentétének tekinthető. **Az <F4> törli az előző parancsnak a megadott karakter első előfordulásáig terjedő részét** (a megadott karaktert már nem törli). Ezután az előző parancsból megmaradt részt az <F1>, <F2> vagy <F3> billentyűk segítségével másolhatjuk át az új parancs-sorba.

Nézzünk meg egy összetett peldát, amelyben az <F4> billentyű is szerepel.

Tegyük fel, hogy az előző DOS parancs ez volt:

COPY A:AUTOEXEC.BAT B:

Ha ezek után az AUTO.BAT nevű file-t akarjuk átmásolni, nyomjuk le először az <F2>, majd az E billentyűket. Erre átmásoljuk az előző parancsból az első E betűig terjedő részt:

COPY A:AUTO

Most jön a törlés: ki akarjuk hagyni az EXEC szót. Nyomjuk meg az <F4>-et, majd a "." (pont) billentyűt. Ez törli az első parancsból a pontig terjedő részt, de magát a pontot nem. A képernyőn a törlést nem látjuk. Végül át kell másolnunk az előző parancs hátralevő részét (.BAT B:). Ezt a műveletet az <F3> lenyomásával végezzük el. A parancs-sorban már a kész parancsot látjuk:

COPY A:AUTO.BAT B:

Ha most lenyomjuk az <ENTER>-t, a DOS végrehajtja a parancsot.

Az <F5> billentyű lenyomásával azt érjük el, hogy az általunk beírt, de még el nem küldött parancs-sor helyettesíti a DOS memóriájában (input pufferében) az előzőleg beírt parancsot. Az új parancs végrehajtására ekkor még nem kerül sor, az eredmény csak annyi, hogy az <F1>, <F2>, <F3>, <F4>, <Ins>, <Del> billentyűkkel ezek után az új parancs-sort lehet tovább feldolgozni, átszerkeszteni. Az <F5> használatára az <Ins> és a <Del> billentyűk ismertetése után mutatunk példát.

Az <Ins> (insert, beszúrás) és a <Del> (delete, törlés) billentyűk segítségével új karaktereket illeszthetünk be, illetve meglévő karaktereket törölhetünk ki az átszerkesztés alatt álló sorban.

Az <Ins> billentyű lenyomása után a gép "beszúró üzemmódba" tér át, és ettől kezdve a beírt karaktereket beilleszti a sor adott helyére, miközben a tőle jobbra álló

karaktereket mindig jobbra tolja. A beszúró módból az <Ins> ismételt lenyomásával vagy bármely más szerkesztő billentyűvel tudunk kilépni.

A <Del> billentyű minden lenyomása egy-egy karaktert töröl az előző (átszerkesztés alatt álló) **sorból**. Eközben a törölt karaktertől jobbra elhelyezkedő szöveg mindig eggyel balra tolódik.

Az alábbi példában az <F5>, az <Ins> és a <Del> szerepét is bemutatjuk. Tegyük föl, hogy ismét a

COPY A:AUTOEXEC.BAT B:

parancsot szeretnénk kiadni. Gépelési hibák következtében azonban a

COOPY A;AUTOEXEC.BT B:

szöveg áll a parancs-sorban (egy O felesleges, kettőspont helyett pontosvessző van, és egy A hiányzik). Nem lenne érdemes a <Backspace>-szel visszatörölni az első O betűig, sem pedig az <Esc> gombbal törölni az egész sort, mert akkor mindent újra kell gépelni. Ehelyett lenyomjuk az <F5> billentyűt, amelynek hatására a hibás sor végére a DOS a @ jelet írja, és egy új, üres sorba tér át, ahol elkezdhetjük a hibás sor átszerkesztését. Az első két betű hibátlan, ezeket az <F1> kétszeri lenyomásával ismételjük meg:

CO

A következő betű, a második O felesleges, ezért a <Del> billentyű egyszeri lenyomásával töröljük. Ez nem látszik a képernyőn. Ismét az <F1>-et kell használnunk a következő négy hibátlan karakter átmásolására (négyszer lenyomjuk az <F1>-et):

COPY A

A pontosvesszőt nem akarjuk átvenni, helyette lenyomjuk a kettőspont billentyűt. Ez felülírja, helyettesíti az eredeti parancs azonos helyén álló karakterét:

COPY A:

Az AUTOEXEC.B szöveget az <F1> tízszeri lenyomásával másoljuk át:

COPY A:AUTOEXEC.B

A hiányzó A betűt a B és a T közé akarjuk beilleszteni. Ehhez nyomjuk le az <Ins> gombot, majd írjuk le az A betűt. Ezután lépjünk ki a beszúró módból az <Ins> újbóli megnyomásával:

COPY A:AUTOEXEC.BA

A parancs hátralevő részét változtatás nélkül akarjuk megismételni, ez az <F3> funkciója, nyomjuk le:

COPY A:AUTOEXEC.BAT B:

Most már hibátlan a parancs, le lehet nyomni az <ENTER>-t.

Ezzel befejeztük a szerkesztő billentyűk tárgyalását, és áttérünk az egyéb feladatokat ellátó speciális billentyűk ismertetésére.

A <PrtSc> (Print Screen, képernyő nyomtatása) felirat általában a csillag-jel (*) fölött szerepel. Első funkciója ezért a <Shift>-tel együtt aktivizálható. **A <Shift-PrtSc> kombináció lenyomásakor a képernyő tartalma kinyomtatódik a számítógép első printer-adapterére kapcsolt nyomtatón** (amennyiben ez be van kapcsolva). Ha a nyomtató nincs bekapcsolva, hibaüzenetet nem kapunk.

A <Ctrl> és a <PrtSc> együttes lenyomásának más a hatása. Ezzel a kombinációval indíthatjuk el az ún. echo (visszhang) üzemmódot. Echo-módban a printeren minden sor megjelenik, amit a billentyűzeten beírunk, valamint az is, amit a számítógép ír a képernyőre. Az echo-módból a <Ctrl-PrtSc> ismételt megnyomásával léphetünk ki. (Ugyanez a szerepe a <Ctrl-P> kombinációnak is.)

Vigyázzunk a <Ctrl-PrtSc> használatakor, hogy a nyomtató be legyen kapcsolva. Ha nincs bekapcsolva - vagy egyéb hiba lép fel -, a DOS hibaüzenetet ír ki (a szokásos abort/retry/ignore? kérdéssel). Ilyenkor hiába akarjuk a parancsot megszüntetni (abort válasz), mivel a nyomtatás echo be van kapcsolva, újabb hibaüzenetet kapunk - ui. a hibaüzenetet is ki akarja nyomtatni a DOS. Ekkor két dolgot tehetünk: vagy bekapcsoljuk a nyomtatót, vagy ismételten (egyszer!) lenyomjuk a <Ctrl-PrtSc> billentyűket, majd addig válaszolunk i-t (ignore) a hibaüzenetekre, míg azok meg nem szűnnek. A <Ctrl-PrtSc> billentyű grafikus üzemmódban csak akkor működik helyesen, ha előtte a GRAPHICS DOS parancsot kiadtuk.

A <Shift-PrtSc>-t a BIOS – a billentyűzet-megszakítást figyelő rutin –, a <Ctrl-PrtSc>-t a DOS kezeli le. Vigyázzunk, hogy - hacsak nem használunk speciális programot erre - grafikus képernyő-kép esetén ne használjuk a <Shift-PrtSc>-t!

Ha a számítógép túl sok információt ír ki egyszerre, és a felső sorok eltűnnek a képernyőről, mielőtt elolvashatnánk azokat, nyomjuk meg egyszerre a <Ctrl-NumLock> billentyűket. Ez megállítja a képernyőre (és a printerre) irányuló listázást, és időt ad a kiírt adatok megtekintésére. A program működése ezzel nem szakad véglegesen félbe (mint a <Ctrl-Break> használatakor), hanem bármelyik betű vagy számbillentyű lenyomásakor ott folytatódik, ahol abbamaradt. A <Ctrl-NumLock> helyett a <Ctrl-S> billentyűt is megnyomhatjuk, a hatás ugyanaz.

Az <F6> billentyűvel lehet file-vége jelet előállítani. A szövegszerkesztők (editorok) és más programok általában kiteszik a file végére e jelet, de egyes esetekben erről magunknak kell gondoskodnunk (például a COPY parancs használatakor, ha az input a billentyűzetről érkezik). Az <F6>-éval megegyezik a <Ctrl-Z> billentyűk hatása is.

Kevésbé fontos a <Ctrl-ENTER> kombináció, amelynek lenyomásakor az írást a következő sorban folytathatjuk anélkül, hogy az előző sort a DOS végrehajtaná.

Az alábbi táblázat röviden összefoglalja a speciális billentyűk és billentyűkombinációk szerepét a DOS-ban:

Billentyű	Hatása
<Alt-Ctrl-Del>	DOS újraindítása
<Ctrl-Break> vagy <Ctrl-C>	Futó program, parancs megszakítása
<Ctrl-NumLock> vagy <Ctrl-S>	Listázás félbeszakítása
<Backspace>	Utolsó karakter törlése (balra)
<ENTER>	DOS parancs (sor) vége
<Ctrl-ENTER>	Új sorba lép át
<Esc>	Sor törlése (az előző sor nem változik)
<F1>	Egy karakter másolása az előző sorból
<F2>	Másolás megadott karakterig
<F3>	Előző sor ismétlése
<F4>	Törlés megadott karakterig
<F5>	Az előző sort az új sorral helyettesíti
<F6> vagy <Ctrl-Z>	File-vége jel
<Ins>	Beszúró mód be/kikapcsolása
<Del>	Egy karakter törlése
<Shift-PrtSc>	A képernyő tartalmának nyomtatása
<Ctrl-PrtSc> vagy <Ctrl-P>	Nyomtatóra írás (echo) be/kikapcsolása

2.5 Lemez- és file kezelés

2.5.1 A fastruktúrájú tartalomjegyzék

A lemez kapacitásának növekedése azt eredményezte, hogy a lemezen egyre több és több file-t lehetett elhelyezni. A nagyméretű tartalomjegyzék (directory) mind a rendszernek, mind a felhasználónak kényelmetlen. Ezért kialakították a többszintű, ún. fastruktúrájú tartalomjegyzéket, amit néha „könyvtár"-nak is neveznek.

A lemez formázása után a rendszer a lemezen előállít egy tartalomjegyzéket. Ezt szokták rendszer-, vagy főtartalomjegyzéknek, vagy gyökér tartalomjegyzéknek (root directory) is nevezni. Amikor a DOS-t elindítjuk a számítógépen, a gyökér tartalomjegyzékben vagyunk.

A tartalomjegyzék mérete:

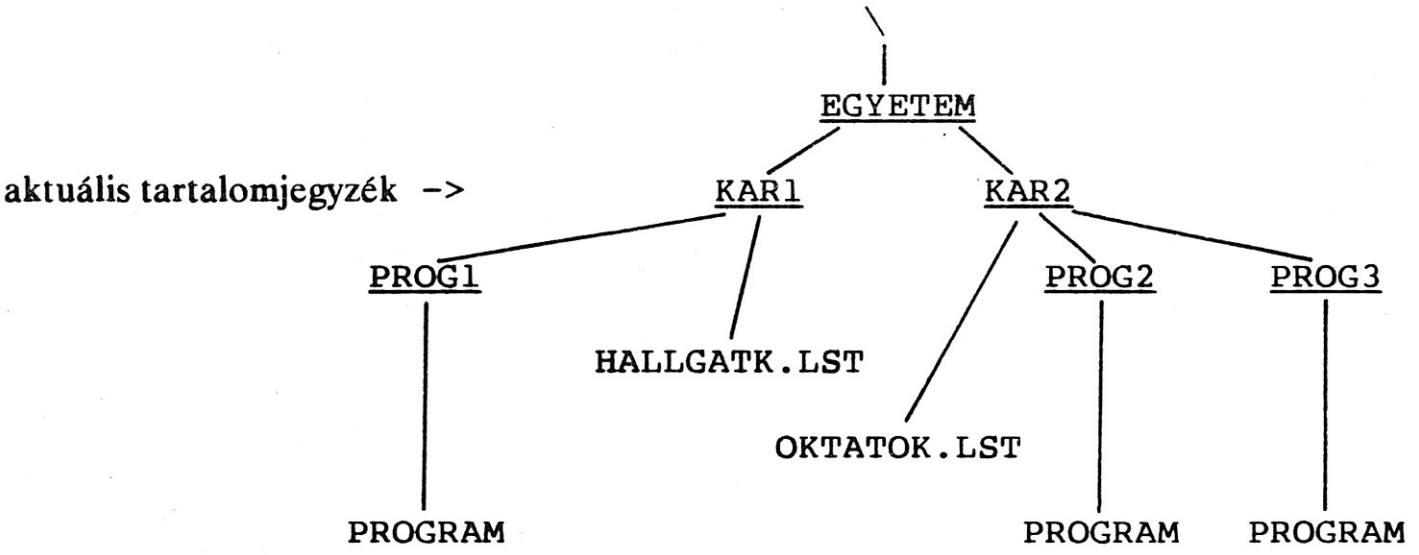
- 64 db bejegyzés 1 oldalú floppy,
- 112 db bejegyzés 2 oldalú floppy,
- 224 db bejegyzés magas kapacitású (high density) floppy esetén,
- fix lemez esetén ez a DOS partíció méretétől függ, de legfeljebb 512 bejegyzés lehet a főtartalomjegyzékben.

A tartalomjegyzékbe nemcsak file-nevek kerülhetnek, hanem további tartalomjegyzéknevek is. Ezek a tartalomjegyzékek szintén tartalmazhatnak további tartalomjegyzékazonosítókat.

Ez utóbbiakat altartalomjegyzéknek nevezzük. Az altartalomjegyzékek mérete nem korlátozott, tetszőleges számú file-név és további altartalomjegyzéknév bejegyzése lehetséges. Méretét csak a rendelkezésre álló lemezterület korlátozza.

A többszintű tartalomjegyzék-rendszert floppy lemezen is ki lehet alakítani, de gyakorlati jelentősége főleg fix lemez esetén van. Normál esetben minden felhasználónak lehet egy vagy több könyvtára (tartalomjegyzéke). Mindenki csak a saját, és a megállapodások szerinti közös tartalomjegyzéket használhatja. Az előny nyilvánvaló.

Egy lehetséges tartalomjegyzék-struktúra a következő:



Jelölés: Az aláhúzott azonosítók altartalomjegyzéket, az aláhúzás nélküliek pedig fileokat jelölnek.

A példában a lemezen több azonos file-név is van. Ez azért lehetséges, mert különböző altartalomjegyzékben találhatók.

A következőkben néhány alapfogalmat ismertetünk a fastrukturájú tartalomjegyzékekkel kapcsolatban.

Aktuális tartalomjegyzék

A rendszernek a következő két információra mindig szüksége van:

- melyik az aktuális meghajtó,
- melyik az aktuális tartalomjegyzék.

A rendszer indítása után az aktuális meghajtón levő lemez gyökér tartalomjegyzékét tekinti aktuális tartalomjegyzéknek. Az aktuális tartalomjegyzék paranccsal megváltoztatható (CHDIR/CD parancs).

File-ok elérése

A rendszernek a file-ok kezeléséhez a következő információkra van szüksége:

- a meghajtó azonosítója,
- tartalomjegyzék megadása, amiben a file található,
- a file azonosítója.

Ha a file az aktuális tartalomjegyzékben található, akkor a tartalomjegyzék azonosítóját nem kell megadni. Ha nem az aktuális tartalomjegyzékben található, akkor meg kell adni a kívánt tartalomjegyzékhez vezető utat, amelyet a tartalomjegyzék-nevek segitségével írunk le.

Az út megadását a fő-, az aktuális- vagy az aktuálist követő tartalomjegyzékkel kezdjük. Az út tartalomjegyzék-nevek sorozatából áll, amelyet \ (backslash) választ el egymástól. Az út a következő formában adható tehát meg:

[\][*altartalomjegyzék-név*\]...

Az út elején álló \ jel azt jelenti, hogy az út nem az aktuális tartalomjegyzéktől, hanem a gyökértől indul. Az út megadásánál két speciális tartalomjegyzék-nevet használhatunk. Az egyik a . (egyetlen pont), ami az aktuális tartalomjegyzéket jelenti. A másik a .. (dupla pont), ami az aktuális tartalomjegyzéket tartalmazó tartalomjegyzék.

(A fenti példában az aktuális tartalomjegyzék a KAR1. A .. tartalomjegyzék így az EGYETEM.)

A file-ok megadása a fentiek alapján a következőképpen történik:

[*d:*][*út*]*file-név*[.*kit*]

A ***d:*** a lemezegység egybetűs azonosítója. Lehetséges értéke a rendszerben levő lemezegységek számától és típusától függ (rendszerint C, D vagy E).

A fenti struktúra esetén a bal szélső PROGRAM file-ra az alábbi módokon hivatkozhatunk:

PROG1\PROGRAM (az aktuális KAR1 tartalomjegyzékből)

vagy

\EGYETEM\KAR1\PROG1\PROGRAM (bárhonnan)

A jobb szélső PROGRAM file az alábbi módon érhető el:

..\KAR2\PROG3\PROGRAM (az aktuális KAR1 tartalomjegyzékből)

vagy

\EGYETEM\KAR2\PROG3\PROGRAM (bárhonnan)

2.5.2 File- és tartalomjegyzék nevek

A file-ok azonosítója két részből áll:

- file-név,
- file-név kiterjesztés (kiterjesztés).

A file-név maximálisan nyolc karakter lehet és a következő jelekből állhat:

- az angol ABC betűi (a-z, A-Z),
- számok: 0-9 és
- a következő speciális jelek: $ # & @ ! % ( ) ' - _

A file-név kiterjesztése mindig a file-név mögött áll, és mindig ponttal (.) kezdődik. A hossza a ponttal együtt maximálisan négy karakter. A kiterjesztés ugyanazokból a jelekből állhat, mint a file-név. A file-név és a hozzá tartozó kiterjesztés együtt alkotja a **file-azonosítót**.

A tartalomjegyzék-név ugyanilyen szabályoknak kell, hogy eleget tegyen. Így a tartalomjegyzék neve is tartalmazhat kiterjesztést, bár ezzel a lehetőséggel ritkán szoktunk élni.

A kiterjesztés a file-azonosítónak nem kötelezően alkalmazandó része. Ha azonban a file létrehozásakor megadtuk, mindig célszerű használnunk. A BASIC például, ha csak filenevet adunk meg, és az hosszabb, mint 8 karakter, a maradékból hozza létre a kiterjesztést.

Például a SZOVEG.TXT és a SZOVEG két különböző file-t azonosít (és mindkettő megengedett azonosító).

Néhány rendszerprogram, fordítóprogram és egyes alkalmazói rendszerek egy adott feladatra meghatározott kiterjesztést használnak. Ezeket felhasználói file-ok azonosításánál nem célszerű felhasználni. A leggyakrabban használt kiterjesztéseket a D. függelékben soroljuk fel.

Ha nem emlékszünk pontosan, vagy nem akarjuk pontosan megadni a file-azonosítót, két speciális karaktert, a csillagot (*) és a kérdőjelet (?) is alkalmazhatjuk a file nevében vagy kiterjesztésében.

A kérdőjel azt jelöli az azonosítóban, hogy helyén bármilyen más megengedett jel állhat. Így minden olyan file kiválasztásra kerül, amelyeknek azonosítója a kérdőjelek helyét kivéve megegyezik.

A csillag (*) a file-azonosítójában azt jelenti, hogy ezen a helyen, valamint a file-név és/vagy a file-kiterjesztés ezt követő pozícióján vagy pozícióin bármilyen megengedett karakter állhat. Ez az egész file-névre, illetve kiterjesztésre vonatkozik. Így a file-név kiterjesztése is - a pont helyét kivéve - tartalmazhat egy csillag karaktert. E két jelet globális név-jelnek hívjuk. Tartalomjegyzék-névben ezeket a karaktereket általában nem használhatjuk!

A globális név-karakterek szerepe egyes parancsok esetén más és más lehet. Van olyan eset, amikor az adott file-névvel azonosítható összes file-lal végrehajtódik az adott művelet, van, amikor csak az elsővel.

Példák:

A DIR parancs a lemez tartalomjegyzékét írja ki. Példánkban a lemez a következő file-okat tartalmazza:

COMMAND.COM
AUTOEXEC.BAT
ANSI.SYS
FORMAT.COM
CHKDSK.COM
SYS.COM
DISKCOPY.COM
DISKCOMP.COM
COMP.COM
EDLIN.COM
MODE.COM
FDISK.COM
BACKUP.COM
RESTORE.COM
PRINT.COM
RECOVER.COM
ASSIGN.COM
TREE.COM
GRAPHICS.COM
SORT.EXE
FIND.EXE
MORE.COM
BASIC.COM
BASICA.COM
KEYBUK.COM
KEYBFR.COM
KEYBSP.COM
KEYBIT.COM
KBDOS.BAS
KBPGM.BAS
WTDATIM.COM
GRAFTABL.COM

A DIR *.EXE és a DIR ????????.EXE parancsok hatása azonos. Ha a parancsok egyikét beadjuk, akkor a következő file-azonosítók jelennek meg a képernyőn:

SORT.EXE
FIND.EXE

A DIR C????????.* és a DIR C*.* parancsok hatása azonos. Ha a parancsok egyikét beadjuk, akkor a következő file-azonosítok jelennek meg a képernyőn:

COMMAND.COM
CHDSK.COM
COMP.COM

A DIR *.* parancs pedig valamennyi file-azonosítót kiírja.

2.5.3 Aktuális lemezegység

Az éppen aktuális lemezegység azonosítóját nem kell a file-névben szerepeltetni. Ez egy sor esetben kényelmessé teszi a DOS használatát, máskor (pl. FORMAT) használatának kellemetlen hatásai lehetnek. Indításkor az aktuális egység az, amelyikről a rendszert betöltöttük. A DOS lehetőséget biztosít az aktuális meghajtó megváltoztatására. Erre a d: parancs szolgál, ahol d valamelyik lemezegység betűjele.

Például:

C > (A kezdeti készenléti jel - a prompt. Tehát a C: jelű fix lemez az aktuális lemezegység.)

C > A: (Az új meghajtó kijelölése)

A > (Az A: lemezegység lett az aktuális meghajtó.)

Megjegyzés:

1. Hogy mi legyen a készenléti jel, a PROMPT paranccsal beállítható. (Lásd a 12. fejezetben a DOS speciális lehetőségeinek ismertetését.)
2. A perifériák hozzárendelésének megváltoztatására is van lehetőség a PC DOS rendszerben. Ez a SUBST (4.16), a JOIN (4.17) vagy az ASSIGN (6.7) paranccsal történhet.
3. Abban az esetben, ha megváltoztatjuk az aktuális lemezegység azonosítóját, akkor minden esetben az utoljára használt tartalomjegyzék lesz az aktuális.

Például ha a rendszert a C: egységről indítottuk el, akkor a következő parancsok hatására az aktuális tartalomjegyzék \TURBO lesz:

```
CD \TURBO
A:
CD \ADATOK
C:
```

Ha ezután újra kiadjuk az A: parancsot, akkor az aktuális tartalomjegyzék ismét \ADATOK lesz.

Egy lemezegység első használatakor a főtartalomjegyzék lesz az aktuális tartalomjegyzék.

2.5.4 Keresési sorrend

A DOS parancsok között két - alapvetően különböző - típus létezik. Az egyik tipust belső, a másikat külső parancsnak hívjuk. A külső parancsok jellemzője, hogy a parancsnévvel azonos .EXE vagy .COM kiterjesztésű file-t a rendszer valamelyik háttértárolón keresi, onnan betölti, majd elindítja.

A rendszer több lemezegységet, azok több altartalomjegyzéket tartalmazhatnak. Alkalmanként pontosan meg kell adni, hogy a parancsot tartalmazó file melyikben is található. Erre több lehetőség is kínálkozik:

a) a parancsot megelőzően beírt elérési út megadásával:

[d:][út] parancs paraméterek

b) A PATH parancs (4.5) használatával.

Ha nem élünk az a), illetve b) lehetőségekkel, akkor a DOS a parancs nevével azonos (.EXE vagy .COM végződésű) file-t csak az aktuális lemezegység éppen használt (aktuális) tartalomjegyzékében keresi.

Az a) alkalmazása esetén a DOS a *d*: lemezegység *"út"* által kijelölt tartalomjegyzékében keresi a parancsnak megfelelő file-t. Ha a *d*: elmarad, akkor az aktuális lemezegység egyik tartalomjegyzékéről van szó. Ha az út megadása marad el, akkor a *d*: lemezegység utoljára használt tartalomjegyzékében fogja a DOS a file-t keresni. Ha a lemezegységet eddig még nem használtuk, ez a főtartalomjegyzék (root directory) lesz.

A PATH parancsban - b) eset - lehetőség van több elérési út megadására is. Ennek egyedül az szab határt, hogy egy DOS parancs hossza maximum 128 karakter lehet.

A B:\DOS\FORMAT parancs hatására a rendszer a B: lemezegység \DOS altartalomjegyzékében keresi először a FORMAT parancsnak megfelelő file-t. Ha a

PATH C:\WORK; D:\ADATOK

parancsot adtuk ki, akkor a rendszer a C: lemezegység \WORK, illetve a D: lemezegység \ADATOK altartalomjegyzékeiben keresi a kiadott parancsnak megfelelő file-t.

A fenti a)-b) sorrend egyben keresési sorrend is. A DOS először a parancs előtt álló [d:][út]-nak megfelelő, vagy ennek hiányában az aktuális lemezegység aktuális tartalomjegyzékében keresi a parancsnak megfelelő file-t, majd, ha itt nem találja, a PATH-ban megadott elérési utakat veszi sorba. Ha egyikben sincs, akkor hibajelzést kapunk.

A belső parancsok előtt nem használhatjuk a [d:][út] megadását. Ennek oka, hogy pl. DIR.EXE vagy DIR.COM nevű file nincs a lemezen.

2.6 A parancsok szintaxisa és a leírásuknál alkalmazott jelölésrendszer

A következő fejezettől kezdve konkrét DOS parancsokat ismertetünk. Elöljáróban megemlítünk néhány szabályt, amely a parancsok általános formájára vonatkozik, és bemutatjuk a parancsok leírásánál használt jelölésrendszert is.

A DOS parancsokat kis- és nagybetűkkel is írhatjuk. A könyv példáiban csak nagybetűket látunk, de ez az írásmód nem kötelező.

A billentyűzetről csak akkor adhatunk ki végrehajtandó DOS parancsokat, amikor a DOS prompt-ját (felszólító jelét) látjuk (pl. A> vagy C>). A parancs beirásának végén nyomjuk meg az <ENTER> billentyűt.

A DOS belső parancsait bármikor kiadhatjuk, de a külső parancsokat csak akkor, amikor az adott parancs programja a DOS számára elérhető (az aktuális, ill. a megnevezett

meghajtó aktuális vagy pontosan megnevezett tartalomjegyzékében, vagy a PATH parancsban kijelölt úton található).

A DOS parancsok általános formája:

kulcsszó paraméterek

A *kulcsszó* határozza meg, milyen műveletet végezzen a DOS (például egy file törlését). A *paraméter*(ek) adja(ák) meg általában a művelet tárgyát (pl. a törlendő file nevét).

A *kulcsszót* a könyvben mindig vastag nagybetűvel írjuk, ezt betűről-betűre pontosan kell lemásolni. Nem minden kulcsszóhoz tartoznak paraméterek. A kulcsszó és az első paraméter között, valamint az egyes paraméterek között egy vagy több szóközt (space) kell hagynunk. (A szóköz helyett más elválasztó jelet: vesszőt, pontosvesszőt, egyenlőségjelet vagy a <TAB> billentyűt is használhatjuk. A könyvben - a jobb olvashatóság kedvéért - szóközt hagytunk az egyes elemek között.)

Előfordulhat, hogy egy parancs több paramétere közül nem mindegyik megadása kötelező. Ha pl. három paraméter közül csak az elsőt és a harmadikat akarjuk kiírni, akkor a szóköztől eltérő elválasztó jelet kell használnunk (pl. vesszőt). A hiányzó paramétert két egymás utáni vessző jelöli:

PARANCS PAR1,,PAR3

A kulcsszón kívül is mindent, amit itt vastag nagybetűvel adunk meg, változtatás nélkül kell beírni, ugyanez vonatkozik a speciális jelekre: pontokra, vesszőkre, csillagokra stb. is.

Amit a könyvben dőlt kisbetűvel jelölünk, azt nem szó szerint, hanem értelemszerűen, az aktuális értékkel helyettesítve kell beírni. Ha a parancsban a *"file-név"* szó szerepel, akkor ehelyett annak a file-nak a nevét kell beírni, amelyikre a parancsot alkalmazni akarjuk.

A szögletes zárójelek ([,]) között álló szövegeket nem kötelező beírni. A parancs leírásából kiderül, mi történik, ha az adott elemet megadjuk, ill. elhagyjuk. A szögletes zárójelek csak jelzések, a parancsok beírásakor ne használjuk azokat! (Vannak olyan esetek, amikor a szögletes zárójelek a parancs részei, s be kell azokat írni. Ezeket mindig külön megemlítjük.)

A függőleges vonallal elválasztott szövegrészek közül választanunk kell, és csak az egyiket kell beírnunk (mindkettőt nem szabad). A függőleges vonal sem része a parancsnak.

A három pont (...) azt jelenti, hogy az előtte álló elemet tetszés szerinti számban meg lehet ismételni.

A számítógép speciális billentyűinek nevét kisebb/nagyobb jelek közé zárva közöljük a szövegben (pl. <ENTER>, <F1> stb.). Ha tehát az <F1> jelöléssel találkozunk, ez azt jelenti, hogy nyomjuk le az F1 feliratú billentyűt, és nem azt, hogy az F és az 1 billentyűket használjuk.

3. DOS: file-kezelő parancsok

3.1 COPY - másolás

A parancs lehetővé teszi

- a file-ok másolását egy másik lemezre a file azonosítójának a megváltoztatása nélkül,
- a file-ok mozgatását ugyanazon a lemezen a file azonosítójának a megváltoztatásával (duplikálás),
- több file összemásolását egyetlenné
- a billentyűzetről, mint a rendszer input perifériájáról, adatok felvitelét a lemezre.

Verzió: 2.11-3.30
A parancs típusa: Belső
A parancs formája:

1. **COPY** [/A][/B][*d:*][*út*]*file-név*[*.kit*] [/A][/B][*d:*][*út*][*file-név*][*.kit*][/V]

azaz:

COPY [*paraméterek*] *forrás file-azonosító* [*paraméterek*] *cél file-azonosító*[/V]

2. **COPY** [/A][/B][*d:*][*út*] *forrás file-név 1* [*.kit*] +
[/A][/B][*d:*][*út*] *forrás file-név 2* [*.kit*] +
...
[/A][/B][*d:*][*út*] *cél file-név* [*.kit*][/V]

A paraméterek jelentése:

A /A és a /B paraméter a **másolás módját adja meg**. Minden paraméter, amit a parancssorba írunk, a következő /A vagy /B paraméterig valamennyi file-azonosítóra vonatkozik. (A /A és /B paraméterek csak a 3.x verziókban használhatók.) Ezeknek a paramétereknek jelentése a következő:

Jelentése a forrás file-ra:

/A a program a file-t ASCII-kódú (szöveges) file-ként kezeli. A file-vége jelig (EOF) másolja a file-t, de azt már nem. (Az EOF <Ctrl-Z>, azaz a hexadecimális 1A.) Tehát több forrás file-t is össze lehet fűzni.

/B a program az egész file-t másolja a tartalomjegyzékben lévő file-méretnek megfelelően.

Jelentése a cél file-okra:

/A A file végére a rendszer kiteszi az EOF jelet.

/B A cél file végére nem kerül EOF.

/V A DOS ellenőrzi a másolat helyességét. Bár csak nagyon ritkán fordulnak elő lemezhibák a másolás során, mégis hasznos lehet ennek a paraméternek a megadása. Ekkor viszont a COPY parancs végrehajtási ideje megnő.

A /V paraméter hatása ugyanaz, mint a VERIFY ON parancsé. Ha a VERIFY ON-t beállítottuk, akkor felesleges a COPY parancsban a /V paraméter megadása. A különbség, hogy a /V csak a COPY végrehajtásának idején él, a VERIFY ON pedig a VERIFY OFF parancs kiadásáig.

Megjegyzés:

1. Ha sem az /A, sem a /B paramétert nem adtuk meg, akkora COPY maga választja meg,hogy melyiket használja. Ha a COPY első formáját használjuk, akkor ennek a paraméternek az alapértelmezése /B, ha file-ok összefűzését adtuk meg, akkor pedig /A.

2. Ha egy fenntartott egységnévről vagy egységnévre másolunk, akkor a /A ASCII-módban kerül végrehajtásra. Az első <Ctrl-Z> jel (EOF) leállítja a másolási folyamatot.

3. Ha egy olyan file-ról készítünk másolatot, amely csak olvasásra (READ ONLY) van kijelölve, akkor a másolat nem kapja meg ezt az attributumot.

4. A globális név-jelek használata (? és *) a paraméterekben megengedett. Ha ezeket használjuk a forrás file-oknál, akkor a másolás során az összes másolt file-nevet a rendszer kiírja a képernyőre.
   Ha a globális név-jeleket a cél file-nál adjuk meg, helyükre a megfelelő forrás filenév karakterei kerülnek. (pl.: COPY *.TXT *.OLD parancsnál az összes .TXT kiterjesztésű file-t átmásoljuk ugyanolyan névvel, de .OLD kiterjesztéssel.)

5. A COPY paraméterében megadott cél file-t, ha nincs olyan azonosítójú file (a megfelelő úton), a COPY hozza létre, ha van, akkor azt kérdés nélkül felülírja.

6. A PROPOS-16-ban a /M opció esetén az eredeti (forrás) file törlődik, /P paraméternél pedig egyenként engedélyezést kér az operációs rendszer, hogy az esetleg létező, azonos nevű file-t felülírja. A /V opció helyett PROPOS-16-ban a /C használatos.

A COPY parancsot három formában szokás használni:

I. Alkalmazás - Másolás ugyanazzal a névvel

Akkor alkalmazzuk, ha a forrás file-azonosítója megegyezik a cél file-éval. Ekkor a parancs formája:

COPY [*d*:][*út*]*file-név*[*.kit*]

vagy

COPY [*d*:][*út*]*file-név*[*.kit*] *d*:[*út*]

Az első formában az aktuális meghajtó aktuális tartalomjegyzékébe kell másolni. A második esetben cél-meghajtót és/vagy tartalomjegyzéket adunk meg. Mivel második file-azonosítót nem adtunk meg, így a cél file is ugyanazt a nevet kapja, ezért szükséges, hogy egy másik meghajtót is specifikáljunk, különben a másolás nem megengedett.

Példa:

1.1. Ha az A: az aktuális meghajtó, és a következő parancsot adjuk ki:

A > COPY C:SZOVEG.TXT

akkor a C: meghajtóról a SZOVEG.TXT azonosítójú file az A: meghajtóra kerül. A SZOVEG.TXT azonosító pedig bekerül az aktuális tartalomjegyzékbe.

1.2. Ha a következő parancsot adjuk ki:

A > COPY *.BAK B:

akkor az A: egységnek azok a file-jai, amelyek az aktuális tartalomjegyzékben vannak és .BAK a file-név kiterjesztésük, a B: meghajtó aktuális tartalomjegyzékébe másolódnak. Az átmásolt file-ok azonosítója megjelenik a képernyőn.

1.3. A következő parancsot a lemez sűrítésére, a file-ok átrendezésére is felhasználhatjuk. Az

A > COPY *.* B:

parancs hatására az A: meghajtó aktuális tartalomjegyzékéből valamennyi file a B: meghajtóra másolódik. Az átmásolt file fizikai blokkjai szekvenciálisan helyezkednek el a lemezen (feltételezve, hogy a B: meghajtóban üres lemez volt). Az átmásolt file-ok azonosítói megjelennek a képernyőn.

II. Alkalmazás - Másolás más névvel

Abban az esetben használjuk, ha a cél file neve más lesz, mint a forrás file-é. A parancs formája következő:

COPY [*út*]*file-név*[*.kit*] [*út*]*file-név*[*.kit*]
vagy
COPY [*út*]*file-név*[*.kit*] *d*:[*út*]*file-név*[*.kit*]

Az első formában egy file-t másoltunk és a másolat nevét megváltoztattuk. Meghajtót nem adtuk meg, ezért az aktuális meghajtót használtuk.

A második formában a file-t átmásoltuk és át is neveztük. Ebben a formában a célmeghajtót is megadtuk. Mivel a file nevét megváltoztattuk, a cél- és a forrásmeghajtónak nem kell különbözőnek lenni. Az aktuális tartalomjegyzék (az a tartalomjegyzék, amiben éppen vagyunk) is lehet ugyanaz.

Példa:

2.1. Ha a következő parancsot adjuk ki:

A > COPY SZOVEG.BAK B:*.TXT

akkor az A: meghajtóról a SZOVEG.BAK azonosítójú file a B: meghajtóra másolódik. A másolat azonosítója a SZOVEG.TXT lesz.

2.2. A másolás során a fenntartott egységnevek is használhatók. Például:

```
COPY CON: ADATOK.TXT
COPY CON: AUX
COPY CON LPT1
COPY ADATA CON:
COPY ADATB AUX:
COPY ADATC LPT2:
COPY AUX: LPT1:
COPY AUX: CON
```

A NUL egységnév is használható. Ha fenntartott egységnevet használunk, akkor nem szükséges a kettőspont (:) megadása. Ennek ellenére - a jobb olvashatóság miatt - célszerű kitenni. Így nem keverhetjük össze file névvel.

2.3. Ebben a példában a COPY parancs segítségével a billentyűzetről viszünk be adatokat és lemezen létrehozunk egy ADATOK.TXT azonosítójú file-t. A parancs a következő:

```
A>COPY CON: ADATOK.TXT
```

Ezután a szöveg beírása következik soronként. A sorok végét az üzenetvégét jelző billentyű, az <ENTER> jelzi.

```
Most új sor következik.
.
.
.
```

Ha befejeztük a szöveg beírását, akkor az <ENTER> után a <Ctrl-Z>-t (és utána <ENTER>-t) kell beírni. Ez írja fel a file-ra a file-vége jelet (EOF).

III. Alkalmazás - File-ok másolása és összefűzése

Akkor használjuk, ha a másolás során a file-okat össze akarjuk fűzni. Ez azt jelenti, hogy egy file-hoz egymás után több file is hozzácsatolható úgy, hogy mindegyik a megelőző file utolsó rekordja után kerül. Az új file file-leíró blokkjában (a file control block-ban) a dátum és az idő a napi dátum és idő, az a mező pedig, amely a file hosszát adja meg, az eredmény file-ra vonatkozik.

COPY [/A][/B][*d*:][*út*]*forrás file-név 1* [*.kit*] +
[/A][/B][*d*:][*út*]*forrás file-név 2* [*.kit*] +
...
[/A][/B][*d*:][*út*] *cél file-név* [*.kit*][/V]

Az összefűzésre kerülő file azonosítókat a COPY parancs paraméterei között fel kell sorolni. Az egyes input file-okat plusz jel (+) választja el egymástól.

Itt a parancsnak a szakasz elején ismertetett második formáját kell használni.

Példa:

3.1. Ha a következő parancsot adjuk ki:

A > COPY SZOVEG1.TXT + SZOVEG2.TXT SZOVEG.TXT

akkor az aktuális meghajtón a SZOVEG.TXT file-t a SZOVEG1.TXT és a SZOVEG2.TXT file-ok összekapcsolásával állítjuk elő.

Megjegyzés:

1. Ha a cél-file azonosítóját nem adjuk meg, akkor a forrás file-ok közül az első azonosítóját tartja meg.

2. A file-ok összefűzésének alapértelmezése a szöveg mód (ASCII kód). Ez azt jelenti, hogy az első EOF jelet (<Ctrl>-Z, azaz hexadecimális 1A) a file végének tekinti.
   A bináris formájú file-ok összefűzésére a /B paramétert kell használnunk. Ez lehetővé teszi, hogy a COPY a teljes file-t átmásolja.
   Az ASCII- és a bináris formájú file-ok összefűzhetők, de akkor a /A és a /B paramétereket értelemszerűen kell használni.
   A /A vagy a /B arra a file-ra vonatkozik, amely előtt áll, és mindaddig vonatkozik az azt követő file-okra, amíg másik /B vagy /A paraméter nem következik.

3. A globális file-név jelek (a * és a ?) a file-azonosítókban megadhatók. A COPY parancs akkor használható a legegyszerűbben, ha a file-ok azonosítóit jól választottuk meg és a globális file-név jelek segítségével tudjuk kijelölni a másolandó file-okat.

4. A file összefűzési eljárás akkor eredményes, ha a COPY legalább egy forrás file-t megtalált. Ha egyetlen file-t sem talált meg, akkor az üzenet a következő:

   0 DATEI(EN) KOPIERT

   illetve

   0 FILE(S) COPIED

3.2 XCOPY - file-ok csoportos másolása

A parancs lehetővé teszi file-ok és altartalomjegyzékben szereplő file-ok különböző szempontok szerinti kiválogatását, majd másolását.

Verzió: 3.20-3.30
A parancs típusa: Külső
A parancs formája:

XCOPY [*d*:][*út*][*file-név*[*.kit*]][*d*:][*út*][*file-név*[*.kit*]][/A][/D][/E][/M][/P][/S][/V][/W]

A paraméterek jelentése:

Az első [*d*:][*út*][*file-név*[*.kit*]] a forrás helyét jelöli ki. Vagy a [*d*:], vagy az [*út*], vagy a [*file-név*[*.kit*]] résznek **szerepelnie kell**. A *file-név*[*.kit*] globális név-karaktereket is tartalmazhat. Ha a *file-név*[*.kit*] elmarad, a teljes tartalomjegyzék tekintendő forrásnak.

A második [*d*:][*út*][*file-név*[*.kit*]] a cél helyét jelöli ki. Ez az a tartalomjegyzék (vagy annak a *file-név* [*.kit*]-tel megadott része), ahová a másolás történik. A *file-név*[*.kit*] tartalmazhat globális név-karaktereket. A file-név megadása csak az átmásolt file-ok átnevezésére szolgál.

/A Csak azok a file-ok kerülnek átmásolásra, amelyek archív attributum bitje beállított. A forrás file-ok archív attributum bitje nem módosul.

/D A dátum megadja, hogy csak az ekkor vagy ezután módosított file-okat kell a parancsnak átmásolnia. A dátumot a SELECT vagy a COUNTRY parancsban megadott formátumnak megfelelően kell megadni. Például így: /D:88-01-23.

/E A parancs azokat az (al)tartalomjegyzékeket is létrehozza a cél tartalomjegyzékben, amelyek (a kiválasztás után) üresek maradnak.

/M Csak azok a file-ok kerülnek átmásolásra, amelyek archív attributum bitje beállított. Az XCOPY a másolás után ezen file-ok archív bitjét törli.

/P Minden egyes kiválasztott file-ról külön is megkérdezi, hogy átmásolja-e. Ha Y-nal válaszolunk, a másolás megtörténik, ha N-nel, akkor nem.

/S A paraméter hatására a forrás tartalomjegyzék altartalomjegyzékét is (teljes mélységben) átmásolja. Az üres, de létező tartalomjegyzékek létrehozására csak a /E paraméter használatával együtt kerül sor.

/V A parancs ellenőrzi a másolás helyességét.

/W A parancs megkezdése előtt az XCOPY a következő üzenetet küldi:

Press any key to begin copying file(s)

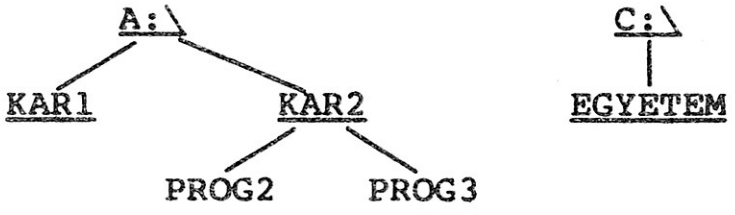
Lehetőség van másik lemez behelyezésére, majd egy tetszőleges billentyű megnyomására megkezdődik a másolás.

Megjegyzés:

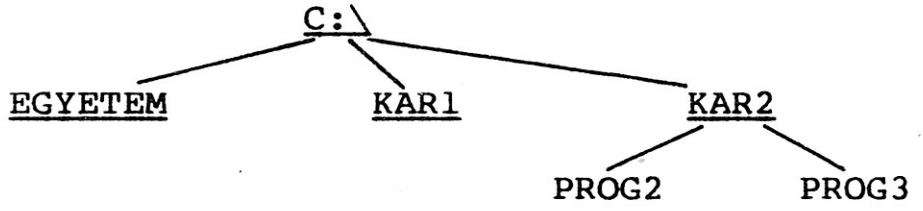
1. A kezdeti tartalomjegyzék és file-név a fő-tartalomjegyzék, illetve a *.* file-név.
2. A másolat file-ok neve is megadható, s ezzel a másolat átnevezhető.
3. Logikai egységekről vagy egységekre (CON, LPT1 stb) nem másolhatunk az XCOPY-val.
4. Az XCOPY a DOS 3.20-nál korábbi verzióiban nem használható.

Példa:

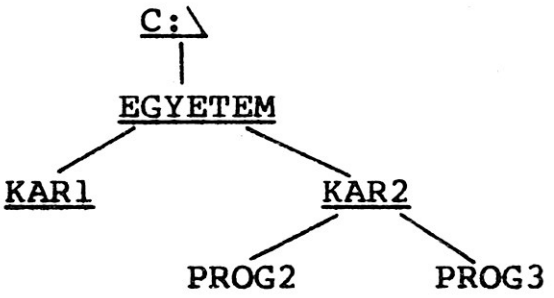
Tegyük fel, hogy az A: és C: lemezek tartalomjegyzéke az alábbi:



Az XCOPY A:\ C:\ /S parancs kiadása után a C: lemez tartalomjegyzéke így alakul:



Ha viszont az XCOPY A:\ C:\EGYETEM /S parancsot adjuk ki a C: egység tartalomjegyzéke így alakul:



3.3 REPLACE - file-ok cseréje

A parancs lehetővé teszi, hogy a célon lévő file-okat a forráson lévő kijelölt file-okkal helyettesítsük.

Verzió: 3.20-3.30
A parancs típusa: Külső
A parancs formája:

REPLACE [*d*:][*út*]*file-név*[*.kit*] [*d*:][*út*][/A][/P][/R][/S][/W]

A [*d*:][*út*]*file-név*[*.kit*] adja meg a forrás file-t, amit a célon helyettesíteni kell /vagy hozzáadni/. A *file-név*[*.kit*] tartalmazhat globális név-karaktereket is. A második [*d*:][*út*] egy cél lemezegységet és annak egy tartalomjegyzékét jelöli ki. Az ebben lévő file-okat kell helyettesíteni, vagy a /A paraméter használata esetén ide kell másolni a forrás file-okat.

A további paraméterek jelentése a következő:

/A A REPLACE a forrásból átmásolja a célra az összes olyan file-t, ami nem létezik a cél tartalomjegyzékben.

/P A parancs minden egyes kiválasztott file után megkérdezi, hogy át akarjuk-e másolni. Y válasz esetén a másolás megtörténik, N válaszra nem.

/R A parancs a cél tartalomjegyzékben lévő, csak-olvasható attributumú file-okat is felülírja.

/S A REPLACE a cél tartalomjegyzék összes altartalomjegyzékét végigkeresi a helyettesítendő file-ok után kutatva.

/W A parancs végrehajtása előtt a meghajtóba kéri a forrás lemezt, pontosabban a következő üzenet jelenik meg:

Press any key to begin replacing file/s/ vagy
Press any key to begin adding file/s/ (/A használatakor)

Példa:

REPLACE C:\MUNKA\PELDA.TXT A:\ /S

A parancs a \MUNKA altartalomjegyzékben lévő PELDA.TXT nevű file-t megkeresi az A: egységen akármelyik altartalomjegyzékben (/S), majd helyettesíti.

Megjegyzés:

1. A /A és /S paraméter nem használható együtt.

2. A parancs az alábbi hibakódokkal tér vissza:

Kód	Hibaüzenet
2	File not found (a file nem létezik)
3	Path not found (az út nem létezik)
5	Access denied (az elérés tilos)
8	Insufficient memory (nincs elegendő memória)
11	Invalid format (érvénytelen formátum)
15	Invalid drive was specified (nem létező lemezegységet adtunk meg)
22	Unknown command (ismeretlen parancs; csak 3.2-nél korábbi verzióban!)

3. A DOS 3.2-nél korábbi verziói nem tartalmazzák ezt a parancsot.

3.4 COMP - összehasonlítás

A COMP (COMPARE, összehasonlítás) parancs lehetővé teszi, hogy file-okat byte-onként összehasonlítsunk. Általában a COMP parancsot a COPY /V parancs helyett használjuk. Meggyőződhetünk vele arról, hogy a másolás művelete sikeres volt-e.

Verzió: 2.11 - 3.30
A parancs típusa: Külső
A parancs formája:

COMP [*d*:][*út*][*file-név*[.*kit*]] [[*d*:][*út*][*file-név*[.*kit*]]

A paraméterek jelentése:

A COMP paranccsal két file hasonlítható össze. Az első parancs paraméter az elsődleges file-t, a második pedig a másodlagost határozza meg. Az összehasonlításra kerülő file-ok azonos lemezen is lehetnek.

A globális név-jelek használata mindkét file-névben megengedett.

Ha a parancsnál nem, vagy hiányosan adunk meg paramétert, akkor a COMP üzenetet küld, és kéri a hiányzó paraméter megadását.

A COMP működése

A parancs először a megadott két file méretét hasonlítja össze. A file-ok méretét a tartalomjegyzékből olvassa ki.

Ha a két méret különbözik, akkor üzenetet küld, és befejezi működését.

Abban az esetben, ha a két méret megegyezik, elkezdi a file-ok elejétől byte-onként az összehasonlítást. Ha a két byte különbözik, akkor hibaüzenetet kapunk. A hibaüzenet tartalmazza a byte-ok címét a file elejéhez viszonyítva és az eltérő két információt hexadecimális formában.

Például:

Compare error at offset 17A
File 1 = 6E
File 2 = 6D

Tíz hibás összehasonlítás után a COMP nem folytatja az összehasonlítást. Befejezi a tevékenységét, és erről a következő üzenetet küldi:

10 Mismatches - ending compare
Compare more files (Y/N)?

Ha a válasz N, a COMP befejezi működését.

A COMP a file-ok összehasonlítását úgy fejezi be, hogy a file-vég-jel meglétét is ellenőrzi (EOF, hexadecimális 1A).

Ha ezt nem találta meg, akkor erről is üzenetet küld. Ez általában bináris file-oknál fordul elő, melyek végét - a szöveg-file-okkal ellentétben - nem jelzi külön EOF byte. Az üzenet bináris file-ok összehasonlítása esetén figyelmen kívül hagyható. Ha az összehasonlításnál nem talált hibát, akkor a következő üzenetet kapjuk:

Files compare ok

Ha a COMP a file-ok összehasonlítását befejezte, üzenetet küld, hogy további file-ok összehasonlítását el kell-e végeznie. Az üzenet a következő:

Compare more files (Y/N)?

Ha a válaszunk N, akkor a COMP befejezi működését. Ha viszont Y a válasz, akkor a COMP megkérdezi az elsődleges és a másodlagos file azonosítóit.

Például:

Compare more files (Y/N)?y
Enter primary file name
autoexec.bat

Enter 2nd file name or drive id
autoexec.1

Példa:

1. Ha például a következő parancsot adjuk meg:

C > COMP A:*.TXT B:*.BAK

ez azt eredményezi, hogy az A: meghajtón lévő valamennyi .TXT kiterjesztésű file összehasonlításra kerül a B: meghajtón lévő azonos nevű, de .BAK kiterjesztésű file-lal.

Az összehasonlításra kerülő file-ok azonosítóit kijelzi.

2. Ha két tartalomjegyzék file-jait szeretnénk összehasonlítani, akkor a következőket kell megadni:

A > COMP C:\BEJEGYZ1 C:\BEJEGYZ2

Ez azt eredményezi, hogy a C:meghajtó BEJEGYZ1 tartalomjegyzékének valamennyi file-ját összehasonlítja a BEJEGYZ2 tartalomjegyzék megfelelő file-jával.

Ha a COMP nem talál egyetlen megegyező azonosítójú file-t sem, akkor ismételten bekéri az elsődleges és másodlagos paramétereket.

Megjegyzés:

1. Mindkét file-nak lehet ugyanaz az azonosítója és még az elérési útja is, feltéve, hogy különböző lemezen találhatók.

2. A COMP parancs nem vár arra, hogy az összehasonlításra kerülő file-okat tartalmazó lemezeket elhelyezzük a meghajtókon. Ezért a COMP kulcsszó megadása után várjuk meg, míg a program a file nevére rákérdez, és csak ezután cseréljük le a lemezt.

3. A PROPOS-16 operációs rendszerben a COMP parancson kívül még a következő parancsok végzik a file-ok összehasonlítását:

CMP [d:][út]file-név[.kit] [d:][út]file-név[.kit][/L][/S]

Ha a /L opciót adjuk meg, akkor a két file valamennyi eltérő byte-ját kilistázza hexadecimális alakban.

Ha a /S opciót írjuk be, akkor csak a visszatérési kódot állítja be a következőképpen:

0, ha a két file megegyezik;
1, ha a két file különböző;
2, ha a file-ok egyikét nem lehet megnyitni.

Ha nem adunk meg opciót, akkor csak az első nem egyező byte-okat írja ki.

COMP [d:][út]file-név[.kit] [d:][út]file-név[.kit][/123]

Két karakteres file-t hasonlít össze soronként. A következő formában küldi az üzenetet:

1. oszlop: csak az elsőként megadott file-ban szereplő sorok;
2. oszlop: csak a második file-ban szereplő sorok;

3. oszlop: azok a sorok, amelyek mindkét file-ban megtalálhatók.

Az opciók megadásával lehet kiválasztani, hogy mely oszlopokat írja ki az outputra. Az opciók tetszés szerinti kombinációban használhatók. Például érvényesek a következő opciók: /12 és /13 vagy /1 /2 /3.

Ha nem adunk meg opciót, akkor nem jelenik meg üzenet. A file-oknak nem kell rendezettnek lenni, és méretük is különbözhet.

4. PROPOS-16 operációs rendszerben további parancsok segítik a karakteres file-ok kezelését.

A DIFF parancs közli, hogy melyek azok a változtatások, amelyek ahhoz szükségesek, hogy az elsőként megadott file megegyezzen a másodikként megadottal. A szükséges változtatásokat soronként közli.

A formája:

DIFF [d:][út]file-név[.kit] [d:][út]file-név[.kit][/E][/B]

Az opciók jelentése:

Alapértelmezés, hogy az első file-ra vonatkozó sorszámmal közli a szükséges változtatást:

I - INSERT (Beszúrás a megadott sorszám elé.)
D - DELETE (Törlés).

A /E hatására a változtatások listája a sorszerkesztő (EDIT) szintaktikájában készül. Ez közvetlenül a sorszerkesztő inputjára is adható.

DIFF ADAT1 ADAT2 /E: EDIT JAVÍTOTT.TXT

A /B opció törli a sorok kezdetén lévő üres (blank) karaktereket és a tabulátor karaktert is.

Gazdaságosan használható programfejlesztésnél, mivel a forrás file-nál jóval rövidebb lesz a DIFF file, az üres karaktereket töröljük.

Ha a DIFF parancs nem tudja megnyitni valamelyik file-t, akkor hibaüzenetet küld. File-csoportok megadása nem lehetséges.

3.5 ERASE (DEL) - törlés

A parancs lehetővé teszi megadott file-ok törlését a tartalomjegyzékből.

Verzió: 2.11 - 3.30
A parancs típusa: Belső
A parancs formája:

ERASE [*d*:][*út*][*file-név*[*.kit*]]
DEL [*d*:][*út*][*file-név*[*.kit*]]

Megjegyzés:

1. Mindkét parancsnál vagy a file-névnek, vagy az útmegadásnak szerepelnie kell.

2. Az ERASE szinonímája a DEL kulcsszó.

3. Globális név-jelek is használhatók. Ha az aktuális tartalomjegyzékből összes file-t törölni akarjuk, akkor a

   DEL [d:]*.*

   parancsot kell beadni.

   Ha ezt a kijelölést használjuk, akkor a végrehajtás előtt a rendszer megkérdezi, hogy valóban minden file-t törölni szeretnénk-e:

   A > del *.*
   Are you sure (Y/N)?

   Biztos benne? (I/N).

   Csak akkor történik meg a törlés, ha Y-t válaszolunk.

   Az IBMBIO.COM és az IBMDOS.COM file-okat és általában a rejtett (hidden) vagy rendszer (SYS) attribútum file-okat, valamint az altartalomjegyzékeket nem lehet ezzel a paranccsal törölni.

4. A szerkesztő-programok egy része a szöveg-file új változatának lemezre írásával egyidőben a régi változatot .BAK kiterjesztésűre változtatja. Ezek egy idő után feleslegessé válnak. Ezeket törölhetjük a C > DEL *.BAK parancs segítségével.

Példa:

1. Ha a következőt írjuk be:

   C > DEL A:PROGRAM.BAS

a DOS törli az A: egységen levő PROGRAM.BAS programot.

2. Ha a következőt írjuk be:

   C > DEL \MYDIR

és \MYDIR egy altartalomjegyzék, akkor a DOS visszakérdezés után törli az összes, a \MYDIR-ben levő file-t.

3.6 RENAME - átnevezés

A parancs file-ok átnevezésére szolgál.

Verzió: 2.11 - 3.30
A parancs típusa: Belső
A parancs formája:

REN[AME] [*d*:][*út*]*file-név*[*.kit*] *file-név*[*.kit*]

A parancs működése:

Az első paraméterben megadott file azonosítóját a második paraméterben megadottra változtatja. A file-név továbbra is ugyanabban a tartalomjegyzékben marad.

Megjegyzés:
1. Az útmegadás csak az első paraméternél használható.
2. A paraméterekben a globális név-jelek is használhatók.

Példa:

Ha a következőket adjuk meg:

REN B:\ALTART1\PROG1.TXT PROG2.TXT

akkor az ALTART1 azonosítójú (al)tartalomjegyzékben lévő PROG1.TXT file-t átnevezi, és a PROG2.TXT az új file-név, ami szintén az ALTART1 altartalomjegyzékben van. Ha a

REN A:*.TXT *.LEV

parancsot adjuk ki, az A: lemez aktuális tartalomjegyzékében lévő összes .TXT kiterjesztésű file kiterjesztését .LEV-re változtatjuk.

3.7 RECOVER - helyreállítás

A parancs lehetővé teszi, hogy a meghibásodott szektorokban található file-okat a hibás területen kívül ismételten előállítsuk.

Verzió: 2.11-3.30
A parancs típusa: Külső
A parancs formája:

RECOVER [*d*:][*út*]*file-név*[*.kit*]

vagy

RECOVER *d*:

A paraméterek jelentése:

Az első formában csak egy file-t fog a rendszer ismételten előállítani. A második formáját akkor használjuk, ha a tartalomjegyzék sérült meg, ekkor a rendszer a lemez valamennyi file-ját ismételten előállítja.

A RECOVER abban az esetben, ha egy file ismételt előállítására használjuk, akkor a file-t szektorról szektorra átmásolja, a hibás részeket átlépi. A file-név nem változik.

Ha a tartalomjegyzék hibás, akkor valamennyi file-t ismételten előállítja, mivel nem tudja megállapítani, hogy melyik bejegyzés a hibás. Ez a file-leíró táblázat (FAT) alapján történik. A file-okat a következő névvel látja el:

FILEnnnn.REC

ahol nnnn a sorszámot jelöli, kezdőértéke: 0001.

Példa:

Ha egy lemez tartalomjegyzéke megsérült, és szeretnénk a tartalmát ismételten előállítani a B: meghajtón, akkor ezt a következő formában tehetjük:

RECOVER B:

A parancs beadása után a rendszer a lemez valamennyi file-ját ismételten előállítja. Az új file-nevek a következők:

FILE0001.REC
FILE0002.REC
.
.
.

Megjegyzés:

1. Ha a globális név-jeleket használjuk, a RECOVER akkor is csak egy file helyreállítását végzi el.
2. Szöveges file-ok ismételt előállítása után rendszerint még egy szövegszerkesztésre is szükség van. A file végéről a felesleges adatokat célszerű törölni.
3. Az első formánál, ha a file futtatható program (.COM illetve .EXE) volt, a hibás szektorok kihagyása miatt a visszanyert alak általában már használhatatlan programot eredményez!
4. Vigyázzunk! Ha a második formát használjuk, a parancs a lemezen levő - jó - file-okat is átnevezi FILExxxx.REC-re. Emiatt a parancs kiadása előtt mindenképpen mentsük el a még használható file-okat!
5. Vigyázat! A parancs a lemezen levő altartalomjegyzékeket egy-egy file-ba sűríti (a FAT alapján), így a bennük levő file-ok szétválogatása eléggé kilátástalan munka.

3.8 ATTRIB - File attributumainak beállítása

A parancs lehetővé teszi, hogy beállítsuk vagy lekérdezzük egy file archív vagy csak-olvasható attributum bitjét.

Verzió: 3.0 - 3.30
A parancs típusa: Külső
A parancs formája:

ATTRIB [+R][-R][+A][-A][*d*:][*út*]*file-név*[*.kit*]

A parancs a [d:][út]file-név[.kit] file archív vagy csak-olvasható attributum bitjét állítja be vagy kérdezi le. Ha a +R/-R, illetve +A/-A paraméterek egyikét sem adjuk meg, akkor a parancs kiírja a file archív, illetve csak-olvasható attributum bitjét.

Az egyes paraméterek hatása a következő:

+R a file csak-olvasható lesz;
–R a file csak-olvasható attributum bitje törlődik;
+A a file archív file lesz;
–A a file archív attributum bitje törlődik.

Megjegyzés:

1. Az archív attributum bitnek a BACKUP /M, illetve XCOPY /M parancsok használatakor van jelentősége.
2. A file-név[.kit] megadásakor használhatunk globális név-karaktereket (*, ?).

Példa:

ATTRIB +R -A KON4.TXT

A parancs hatására a KON4.TXT file csak-olvasható lesz, s az archív attributum bit törlődik.

3.9 SHARE - file megosztási környezet kezelése

A parancs betölti a file megosztási környezetet, ami lehetővé teszi a file-ok írásának és olvasásának ellenőrzését. Részletesen erről a 3DH és 3CH sorszámú DOS funkciók tárgyalásakor szólunk a 17. fejezetben.

Verzió: 3.0 - 3.30
Parancs típusa: Külső
Parancs formája:

SHARE [/**F**:*file méret*][/**L**:*zárolások*]

A paraméterek jelentése:

/F file méret megadja az írási/olvasási műveletek ellenőrzéséhez szükséges memória terület nagyságát. Minden megnyitott file 22 byte-ot foglal el. A kezdeti érték 2048.
/L zárolások megadja a használni kívánt zárolások számát. Kezdeti értéke 20.

Megjegyzés:

1. Ha a SHARE parancsot másodszor kísérelјük meg kiadni, akkor "Share Already Installed" üzenetet kapunk.
2. Egyes lemezegységek képesek annak ellenőrzésére, hogy kicseréltük-e benne a lemezt vagy sem. Ha ilyen történik egy írási/olvasási művelet alatt, a SHARE ellenőrzi, hogy jó lemez van-e bent, majd megismétli a műveletet.
3. A SHARE parancs a DOS 3.0-ásnál korábbi verzióiban nem használható.

3.10 TYPE - kiíratás

A parancs lehetővé teszi, hogy a file tartalmát a képernyőre kiírjuk.

Verzió: 2.11 - 3.30
A parancs típusa: Belső
A parancs formája:

TYPE [*d*:][*út*]*file-név*[*.kit*]

A parancs működése:

A megadott file tartalma megjelenik a képernyőn. A karakteres file-ok olvasható formában kerülnek kijelzésre. A bináris file-ok esetén ez a megjelenítés általában nem sok információt ad a file-ról.

A képernyő tartalmát nyomtatóra is kiírathatjuk a <Ctrl-PrtSc> billentyűk lenyomásával. Az outputot a nyomtatóra vagy egy file-ba is átirányíthatjuk. (Ezt a 12. fejezetben ismertetjük "A standard I/O átirányítása" című szakaszban.)

Megjegyzés:

1. A globális név-karakterek nem használhatók. Ha mégis megpróbálkozunk velük, akkor a következő üzenetet kaphatjuk:

   FILE NOT FOUND
   (A file nem található.)

2. A szöveges file-ban lévő tabulátor-karaktereket a TYPE felismeri. A tabulátor után következő karaktert a megfelelő tabulátor-pozícióba helyezi. (A tabulátor pozíciók: 8., 16., 24., 32. oszlop és így tovább.)

3. A PROPOS-16 operációs rendszerben a TYPE parancs paraméterek nélkül is megadható. Ekkor a rendszer a további parancs-paramétereket a standard input perifériáról várja.

4. A file képernyőre listázása megállítható a <Ctrl-S> vagy a <Ctrl-NumLock> billentyű megnyomásával. Ilyen esetben a képernyőről a <Ctrl-PrtSc> használatával másolat készíthető a nyomtatóra. A listázás tetszőleges billentyű megnyomására folytatható. A <Ctrl-C> megnyomása megszakítja a parancs végrehajtását.

Példa:

1. Az AUTOEXEC.BAT file tartalmát például a következő parancs kiadásával olvashatjuk el:

   C>TYPE AUTOEXEC.BAT

2. Ha a file tartalmát ki akarjuk a nyomtatóra íratni, ezt a standard output átirányításával érhetjük el: C>TYPE PROGRAM.PAS >PRN:

A fenti parancs hatására a PROGRAM.BAS file tartalma a nyomtatóra kerül.

Hosszú szövegeket kényelmesebben olvashatunk, ha képernyőnként íratjuk ki a szövegeket. Ezt például a fenti 'PROGRAM.BAS' file-lal a következőképpen tehetjük meg:

C>TYPE PROGRAM.BAS‡ MORE

3.11 PRINT - nyomtatás

A parancs lehetővé teszi, hogy a nyomtatón listákat készítsünk, miközben a számítógép más feladatokat is ellát.

Verzió: 2.11 - 3.30
A parancs típusa: Külső
A parancs formája:

PRINT [/D:*egységnév*][/B:*pufferméret*][/U:*foglalt*][/M:*nyomtat*][/S:*várakozik*]
[/Q:*sorméret*][/C][/T][/P][*d*:][*út*][*file-név*.[*kit*]...]

A PRINT működése

A parancs **spooling** funkciót valósít meg. A parancs első kiadása után a rendszer mérete kb. 3200 byte-tal megnövekszik. Ezen a memóriaterületen, mintegy "BACKGROUND" partícióban működik a PRINT, miközben a számítógépen tovább dolgozhatunk.

A program rendelkezik egy várakozási sorral, amelyben legfeljebb 32 file-nevet tud tárolni. A PRINT paranccsal lehet a várakozási sort kezelni. A sorhoz hozzáadhatók újabb file-nevek, törölhetők belőle, de a teljes várakozási sort is lehet törölni.

A paraméterek jelentése:

/T	hatására a várakozási sor valamennyi eleme törlődik,
/C	segítségével a megadott file-okat töröljük a várakozási sorból,
/P	segítségével új file-neveket fűzhetünk fel a várakozási sorba. (Ez az alapértelmezés.)
/B:*pufferméret*	segítségével a PRINT belső pufferének hosszát adhatjuk meg. Ezt a PRINT 512-nek választja meg. A nagyobb pufferméret hatékonyabbá teszi a parancs használatát.
/U:*foglalt*	azoknak az óraciklusoknak a száma, ameddig a PRINT-nek várakoznia kell, míg a nyomtató elérhető lesz ismét. A PRINT ezt az értéket 1-re állítja be. Az U: foglalt értéknek az 1-255 intervallumba kell esnie.
/M:*nyomtat*	azoknak az óraciklusoknak a száma, amennyi idő alatt a nyomtató ki tud nyomtatni egy karaktert. A PRINT ezt 2-re állítja be; lehetséges értéke az 1-255 intervallumba kell, hogy essen.
/S:*várakozik*	a várakozási időszak számát adja meg. Ha a rendszerben nem állítottuk be a végtelen sokszori kísérletezést, akkor ennyiszer próbálkozik a PRINT, mielőtt úgy dönt, hogy a nyomtató egység használhatatlan. A PRINT ezt az értéket kezdetben 8-ra állítja be. A lehetséges érté-

kek: 1-255.

/Q:*sorméret* a várakozási sor nagyságát adja meg. Alapértelmezése 10, értéke 1 és 32 közötti egész szám lehet.

/D:*egységnév* a PRINT által használandó eszköz nevét adja meg. Ha a parancs első alkalmazásakor nem adjuk meg ezt a paramétert, a PRINT megkérdezi, hogy melyik egységet akarjuk használni. A lehetséges válaszok:

LPT1, LPT2, LPT3, PRN, COM1, COM2, AUX.

Megjegyzés:

1. A /D, /B, /Q, /S, /U és /M paramétereket csak a parancs első hívásakor használhatjuk.

2. A DOS 2.10 és 2.11-es verziójában csak a /C, /T és /P paraméterek használhatók.

3. A PRINT használata mellett más módon (pl. <PrtSc> vagy <Ctrl-P>) nem használható a kijelölt periféria. Ha ezzel kísérletezünk, "out-of-paper" hibaüzenetet kapunk.

4. A parancssorban több file-név is megadható. Az azonosítókban a globális név-jelek is használhatók.

5. Csak az aktuális tartalomjegyzékben szereplő file-neveket lehet egy várakozási sorban megadni.
A várakozási sor megadása után az aktuális tartalomjegyzék változtatható anélkül, hogy a várakozási sort ez érintené.

6. A kinyomtatandó file-okat tartalmazó lemezt mindaddig a megadott meghajtóban kell hagyni, amíg a teljes nyomtatási folyamat be nem fejeződik.

7. Ha a DOS a nyomtatás folyamán a lemez olvasása során hibát észlel, akkor a következő történik:

   - A nyomtatás alatt álló file-t törli a várakozási sorból.
   - Hibaüzenetet küld a nyomtatóra.
   - A nyomtató új lapot kezd és hangjelzést is ad.
   - A várakozási sorban következő file kinyomtatását elkezdi.

8. Ha a /T vagy a /C paramétert nyomtatás alatt álló file esetében adjuk meg, akkor a következő történik:

   - A nyomtató hangjelzést ad.
   - A nyomtatón üzenet jelenik meg a törölt file-(ok)ról.
   - A nyomtató új lapot kezd.
   - Ha a várakozási sorban még van file, akkor annak a nyomtatásával folytatódik a kiírás.

Példa:

1. Tételezzük fel, hogy a PRINT első használatakor a következő parancsot adtuk ki.

```
C>PRINT A:LISTA.TXT
```

A parancs hatására a PRINT a memóriába töltődik, és a következő üzenetet kapjuk:

Name of List device [PRN]:
(a listázó eszköz neve [PRN]:)

Válaszként azt az output eszközt kell megadni, amelyikre a file-okat majd ki akarjuk iratni. Az < ENTER > megnyomása önmagában a PRN: használatát eredményezi.

2. Ha már elindult a nyomtatási folyamat, és a következőt adjuk meg:

A > PRINT A:LIST3.TXT/C A:LIST4.TXT A:LIST5.TXT

akkor a LIST3.TXT törlődik a várakozási sorból, a LIST4.TXT és a LIST5.TXT pedig bekerül a várakozási sorba.

3. Ha a teljes várakozási sort szeretnénk törölni, akkor a következőt kell megadni:

A > PRINT /T

A rendszer ekkor üzenetet is küld, hogy minden file-nevet törölt a várakozási sorból.

3.12 FASTOPEN - file-ok elérésének gyorsítása

A DOS, ha egy olyan file-t használunk, amelynek elérési útja hosszú, a file-műveletek során a teljes elérési utat végigjárja. Ez olyan programok esetén, amelyek gyakran fordulnak a lemezegységhez, sok időt elrabolhat.Ezt gyorsítja a FASTOPEN parancs.

Verzió: 3.30
A parancs típusa: külső
A parancs formája:

FASTOPEN *d*:[=*nnn*]...

A paraméterek jelentése:

d annak a lemezegységnek az azonosítója, amelyiken a gyors elérést be akarjuk vezetni.

nnn megadja azoknak a tartalomjegyzék és file-név bejegyzéseknek a számát, amelyek helyét a DOS a memóriában megjegyezhet. Ha ez a szám nagyobb, mint az általunk használt file-nevekben összesen szereplő tartalomjegyzék és file-nevek száma, akkor a file-ok elérése nagyon meggyorsul. nnn értéke legalább 10,legfeljebb 999 lehet. Alapértelmezése 34.

Példák:

FASTOPEN D:
FASTOPEN C: = 120 D: = 60

4. DOS: lemezkezelési parancsok

A DOS 2.00-ás verziója biztosította először a felhasználónak a többszintű tartalomjegyzékek létrehozását. Az ezt megelőző verziókban sem ez, sem a fix lemez kezelése még nem állt a felhasználók rendelkezésére. Az ebben a fejezetben ismertetésre kerülő parancsok a következő funkciókat látják el:

- többszintű tartalomjegyzék kezelése,
- tartalomjegyzék kijelzése,
- lemez formázása,
- lemez másolása,
- lemezek összehasonlítása,
- lemez ellenőrzése és
- írás utáni ellenőrzés beállítása.

4.1 MKDIR - altartalomjegyzék létrehozása

A parancs lehetővé teszi, hogy egy tartalomjegyzékben létrehozzunk egy altartalomjegyzék bejegyzést.

Verzió: 2.11 - 3.30
A parancs típusa: Belső
A parancs formája:

MKDIR [*d*:]*út*
MD [*d*:]*út*

Megjegyzés:

1. Egy lemezen tetszőleges számú altartalomjegyzék előállítható, számukat csak a lemez kapacitása korlátozza. (A főtartalomjegyzékben azonban csak korlátozott számú - elsődleges - altartalomjegyzék bejegyzés lehet.)
2. Az út maximális hossza nem lehet nagyobb, mint 63 karakter, beleértve a \ (backslash) jeleket is.
3. Ha az út megadásánál \ (backslash) nem szerepel, akkor az altartalomjegyzék bejegyzés az aktuális tartalomjegyzékbe kerül.
4. Egy újonnan létrehozott tartalomjegyzék két bejegyzést tartalmaz. Ezek a . és a .. bejegyzések, amelyek a tartalomjegyzékre magára, illetve az ezt közvetlenül tartalmazó tartalomjegyzékre utalnak.
5. Egy tartalomjegyzéket akkor hívunk üresnek, ha a . és .. bejegyzéseken kívül nem tartalmaz mást. (Megjegyzés: Mivel altartalomjegyzékben lehetnek rejtett file-ok is, nem biztos, hogy ha a DIR parancs csak a . és .. bejegyzéseket mutatja, akkor az altartalomjegyzék üres.)

Példa:

1. Ha a főtartalomjegyzékben akarunk létrehozni egy altartalomjegyzék-bejegyzést (SZINT1), akkor a következőt adhatjuk meg:

MD \SZINT1

Ha további altartalomjegyzéket akarunk létrehozni (SZINT2), akkor a következőt adhatjuk meg:

MD \SZINT1\SZINT2

2. Ha az aktuális tartalomjegyzékbe akarjuk beírni a SZINT4 altartalomjegyzék bejegyzést, akkor a következőt adhatjuk meg:

MD SZINT4

4.2 CHDIR - aktuális tartalomjegyzék kijelölése

A parancs segítségével egy megadott tartalomjegyzék aktuális tartalomjegyzékként kijelölhető.

Verzió: 2.11 - 3.30
A parancs típusa: Belső
A parancs formája:

CHDIR [[*d*:]*út*]
CD [[*d*:]*út*]

Megjegyzés:

1. A DOS minden file-t, amit út kijelölés nélkül adunk meg, az aktuális tartalomjegyzékben keres.
2. Ha csak a CD vagy a CHDIR kulcsszót adjuk meg, akkor az aktuális tartalomjegyzék elérési útját írja ki válaszul a rendszer.

Példa:

1. A 2.5 szakasz elején adott ábra szerinti struktúrában a \PROG1-be a következő úton juthatunk:

CD\EGYETEM\KAR1\PROG1

A parancs kiadása után az aktuális tartalomjegyzék a PROG1 lesz.

2. Ha a következőt adjuk meg:

CHDIR \

akkor az aktuális meghajtó főtartalomjegyzéke lesz az aktuális tartalomjegyzék.

3. Ha az aktuális meghajtón levő aktuális tartalomjegyzékre vagyunk kíváncsiak, akkor a

CD

parancs kiadásával kaphatjuk ezt meg. A

CD D:

parancs kiadására válaszul a D:egység aktuális tartalomjegyzékét írja ki a DOS.

4.3 RMDIR - altartalomjegyzék törlése

A parancs lehetővé teszi, hogy a megadott tartalomjegyzékből altartalomjegyzéket töröljünk.

Verzió: 2.11 - 3.30
A parancs típusa: Belső
A parancs formája:

RMDIR [*d*:]*út*
RD [*d*:]*út*

Megjegyzés:

1. Egy altartalomjegyzék csak akkor törölhető, ha az sem altartalomjegyzék, sem file bejegyzést már nem tartalmaz, ezért az altartalomjegyzékek törlését mindig a legalsó szinten kell kezdeni.
2. A főtartalomjegyzék és az aktuális tartalomjegyzék nem törölhető.
3. Az út megadásában szereplő utolsó tartalomjegyzéknév a törlendő tartalomjegyzék.
4. Ha egy látszólag üres altartalomjegyzéket a parancs nem hajlandó törölni, akkor vagy ez az aktuális (akkor át kell térni másra), vagy rejtett file-okat tartalmaz (melyeket a DIR nem jelez ki). Utóbbi esetben csak speciális programokkal nézhetjük meg, milyen file-ok vannak itt és csak ilyenekkel tudjuk ezeket törölhetővé tenni.

Példa:
1. Ha a következőt adjuk be:

RD \SZINT1\SZINT2\SZINT3

akkor a SZINT2 tartalomjegyzékből a SZINT3 törlődik.

2. Ha a teljes fastruktúrát törölni szeretnénk, akkor azt a következő lépésekben tehetjük:

RD \SZINT1\SZINT2
RD \SZINT1

Természetesen előtte az altartalomjegyzékből a file-okat is törölni kell.

4.4 TREE - tartalomjegyzék struktúrájának kiíratása

A parancs lehetővé teszi, hogy egy lemez összes elérési útját kilistázzuk.

Verzió: 2.11 - 3.30
A parancs típusa: Külső
A parancs formája:

TREE [*d*:][/F]

Megjegyzés:

1. Minden megtalált tartalomjegyzéknél a teljes elérési utat kijelzi és megadja az altartalomjegyzékeinek a listáját is.
2. Ha a /F paramétert megadjuk, akkor az altartalomjegyzékben szereplő valamennyi file-nak az azonosítóját is kijelzi.
3. Az outputot célszerű a nyomtatón is kiíratni, vagy átirányítani. (Lásd 12. fejezet "A standard I/O átirányítása" című alfejezetet.)
4. A PROPOS-16 operációs rendszerben az FSYS parancs végzi el a TREE funkciót. Paraméterezése is azonos.

Példa:

1. A nyomtatóra átirányítva a kiíratást:

TREE C:/F > PRN

2. A lista egy file-ba kerül, amelyet a PRINT paranccsal kiírathatunk. Ekkor a következőt adjuk meg:

TREE C:/F > TARTJEGY.LST

4.5 PATH - bejárási utak kijelölése programhoz

A parancs lehetővé teszi olyan programok elérését is, amelyek az aktuális tartalomjegyzéken kívül találhatók. Segítségével több elérési út is kijelölhető, illetve a kijelölt utak lekérdezhetők.

Verzió: 2.11 - 3.30
A parancs típusa: Belső
A parancs formája:

PATH [[*d*:]*út*[[;[*d*:]*út*]...]]

vagy

PATH;

Megjegyzés:

1. Az útkijelöléseket mindig pontosvessző (;) választja el. Ha a DOS a file-t az aktuális tartalomjegyzékben nem találja, akkor a keresést abban a sorrendben végzi, ahogyan azt a felhasználó a parancsban meghatározta.

   Abban az esetben, ha több azonos nevű file is van az útkijelölések tartalomjegyzékeiben, akkor azt fogja megtalálni, amelyiket a parancs megadásakor előbb jelöltünk ki.

2. Az útkijelölések a következő paranccsal törölhetők:

   PATH ; (a parancsot csak egy pontosvessző követi)

   Ha a rendszert újraindítjuk, a DOS már "nem emlékszik" az útkijelölésekre. (Kivéve arra, ami az AUTOEXEC.BAT-ben szerepel.)

3. Ha az érvényben lévő útkijelöléseket akarjuk kiíratni, akkor a következőt kell beírni:

   PATH

4. Az útkijelölésben lévő helytelen információkat a DOS csak akkor fedezi fel, ha valamely bejárási utat végignézi. A hibás, nem létező utakat átlépi, és a következő útkijelölésnél folytatja a keresést.

5. A PATH parancs segítségével csak a .COM, .EXE és .BAT kiterjesztésű file-okat érhetjük el. Ha ettől eltérő kiterjesztésű file-jaink vannak, akkor azok elérése az APPEND (3.30 és későbbi verziók) parancsot használjuk.

Példa:

1. Egy út kijelölése a következő lehet:

   PATH C:\SZINT1\SZINT2\SZINT3

2. Több út megadására példa az alábbi:

   PATH UTVONAL; EGYETEM; EGYETEM KAR1

4.6 FORMAT - lemez formázása

A parancs lehetővé teszi, hogy a lemezt előkészítsük DOS file-ok befogadására. A parancs ellenőrzi, hogy a lemez nem sérült-e. Felírja az üres tartalomjegyzék-blokkot és inicializálja a file-leíró táblázatot (FAT).

Verzió: 2.11 - 3.30
A parancs típusa: Külső

A parancs formája:

FORMAT [*d*:][/S][/1][/8][/4][/V][/B][/N:*xx*][/T:*yy*]

A paraméterek jelentése:

d: azt a meghajtót adja meg, amelyen a formázandó lemez van.

/S az aktuális meghajtón lévő lemezről az operációs rendszer file-jait az új, most formázandó lemezre másolja. Ha e paramétert használjuk, és az aktuális meghajtón éppen nincs rajta a rendszer, két eset van. Ha az aktuális meghajtó a Winchester, akkor a rendszer felszólít, hogy tegyük az "A" meghajtóba a rendszerlemezt. Ha az aktuális meghajtón cserélhető a lemez, akkor oda kéri a rendszerlemezt.

/1 azt jelöli, hogy a lemezt egyoldalas felhasználásra formázzuk.

/8 azt jelöli, hogy a lemezt sávonként 8 szektorral formázza. Alapértelmezés: 9 szektor/sáv.

/4 használata biztosítja, hogy a nagy-kapacitású lemezegységben (1.2 Mbyte-os) a lemezt normál (1 vagy 2 oldalas) lemezként formázza meg a parancs. Nem használható a /1, illetve /8 paraméterekkel.

/V lehetővé teszi, hogy kötetazonosítót adjunk a lemeznek. Ez a lemez azonosítására szolgál. (PROPOS-16 esetén /L a paraméter.)

/B Egy olyan lemezt, amelyet sávonként 8 szektorosra formázunk, úgy formattál, hogy legyen rajta hely az IBMBIO.COM és az IBMDOS.COM rendszerprogramoknak. A /B paraméter sem ezeket, sem a parancsfeldolgozó rendszerprogramot (command processor) nem teszi rá a lemezre, csak helyet biztosít arra, hogy majd a DOS bármely változatát erre a lemezre a SYS paranccsal rátehessük.

/N:*xx* megadja az egy sávra írandó szektorok számát.

/T:*yy* megadja a megformázandó sávok számát.

Megjegyzés:

1. A /1, a /4 és a /8 paraméter fix lemez esetén nem használható.

2. A /V és a /S paramétereket nem lehet a /B paraméterrel együtt használni.

3. A /S esetén a következő file-ok másolódnak a lemezre:

   IBMBIO.COM
   IBMDOS.COM
   COMMAND.COM (parancsfeldolgozó rendszerprogram).

4. Az alábbi táblázat összefoglalja, hogy adott típusú lemezegység esetén ilyen paraméter használható:

Típus	**Megengedett paraméter**						
fix lemez	/S	/V					
1.2 Mbyte	/S	/V	/4				
720 Kb/144 Mb	/S	/V	/N	/T			
320/360 Kbyte	/S	/V	/1	/8	/B		
160/180 Kbyte	/S	/V	/1	/8	/B	/N	/T

5. A formázás során hibásnak talált lemezterületeket a program fenntartott sávoknak jelöli meg, így megakadályozza, hogy ezekre a területekre file-ok kerüljenek.

6. Ha a /V paramétert megadtuk, akkor a parancs bekéri a kötetazonosítót, ez maximálisan 11 karakter lehet. Azok a karakterek használhatók, amelyek a file-névben is előfordulhatnak.

7. A fix lemezeket többnyire már a gyártók formázzák. A formázás során a DOS partíció minden szektora ellenőrzésre kerül, de újraformázás nem történik, így gyakran előfordul, hogy néhány kbyte hibás, ettől ne ijedjünk meg.

8. Azok a floppy lemezek, amelyeket a gyártó egyoldalasként jelölt meg, többnyire megformázhatók kétoldalasra is.
Az egyoldalas lemezek egy- és kétoldalas meghajtón is használhatók (de célszerűen egyoldalasra formázva).
A kétoldalas lemezek egyoldalas meghajtón is használhatók (természetesen csak egyik oldalukat használhatjuk!).

9. A FORMAT a következő üzeneteket küldi:

   - a lemez teljes tárolókapacitása,
   - a hibás terület mérete,
   - a rendszer file-oknak fenntartott terület,
   - a rendelkezésre álló szabad terület nagysága.

10. A DOS 3.20 és későbbi verziók a fix lemez véletlen megformázása ellen bizonyos védelmet tartalmaznak, nevezetesen a DOS megkérdezi a fix lemez (jelenlegi) kötetazonosítóját. Ha ezt nem tudjuk megadni, a formázás elmarad. Ha nem tudjuk a kötetazonosítót, akkor a FORMAT parancs kiadása előtt adjuk ki a VOL parancsot!

11. A parancs ismételt beírása nélkül több lemez azonos módon megformázható.

12. Az /N és a /T paramétereket egyszerre kell használnunk.

13. A parancs a következő kódokkal tér vissza:

Kód	Jelentés
0	Sikeres befejezés (az utolsó formázásé)
1-2	Nem használt
3	< Ctrl-Break > -kel megállítva
4	Hiba történt
5	Fix lemez formázásakor N-nel válaszoltunk.

14. A FORMAT parancsot ne adjuk ki olyan meghajtóra, amire a SUBST vagy JOIN parancsokban hivatkozunk. A FORMAT hálózati lemezegységre sem adható ki.

Példa:

Ha a konfigurációnk egyetlen floppy meghajtót tartalmaz, akkor a lemez formázása a következő módon történhet:

A > FORMAT A:

Ezután a rendszer a következő üzenetet küldi:

Insert new diskette for drive A:
and strike any key when ready
(Tegye az új lemezt az A: meghajtóba,
majd nyomjon meg egy billentyűt.) (DOS 2.xx)

illetve

Insert new diskette for drive A:
and press ENTER when ready
(Tegye az új lemezt az A: meghajtóba,
majd nyomja meg az ENTER billentyűt.)

Ezt követően a rendszer elkezdi a lemez formázását, és a következő üzenetet küldi:

Formatting ...
(Formázás folyik)

A DOS 3.20 és későbbi verziói egy Head:xxx Cylinder:yyy üzenetet is küldenek, amely segítségével nyomon követhető a formázás előrehaladása.

A formázás befejezésekor a következő üzenetet küldi:

Formatting... Format complete
362496 bytes total disk space
362496 bytes available on disk
(Formázást befejezte.
362496 byte a teljes lemezkapacitás.
362496 byte a szabad terület a lemezen.)

Ezután a következő üzenetet küldi:

Format another (Y/N)?
(Kell-e még másik lemezt formázni? (I/N)?

Erre az Y választ adjuk, ha igen, egyébként N-et.

Fix lemez formázásakor a program megkérdezi, valóban formázni akarjuk-e Winchesterünket:

WARNING! ALL DATA ON NON-REMOVABLE DISK DRIVE d: WILL BE LOST!
Proceed with Format (Y/N)?

(Figyelem! A fix lemezen levő összes adatállomány el fog veszni!
Valóban el akarja indítani a formázást?)

Amennyiben Y-t válaszolunk, a fix lemez formázása elkezdődik. Ettől a pillanattól kezdve hiába állítjuk le vagy kapcsoljuk ki a gépet, a file-ok már elvesztek!

4.7 SELECT - új rendszerlemez előállítás adott billentyűzet rutinnal, dátum és idő formátummal

A parancs lehetővé teszi, hogy egy új DOS lemezt állítsunk elő megadott billentyűzet rutinnal, dátum és idő formátummal.

Verzió: 3.00 - 3.30
A parancs típusa: Külső
A parancs formája:

SELECT [[**A:** | **B:**] *d*:[*út*]] *xxx yy*

A paraméterek jelentése:

A: vagy B: a forrásmeghajtót jelenti. Csak ezeket az egységeket használhatjuk. Ha egyiket sem adjuk meg, akkor az A: egységet használja a parancs. A d:[út] a cél tartalomjegyzéket adja meg, ide másolja a DOS a parancs file-okat. Ha nem adjuk meg, akkor a főtartalomjegyzékbe kerülnek. xxx az ország kódját adja meg. yy a billentyűzet rutin kódját adja. Lehetséges értékeik a következők:

Ország	Kód	Billentyűzet rutin
Ausztrália	061	*
Belgium	032	*
Kanadai francia	002	*
Dánia	045	*
Finnország	358	*
Franciaország	033	FR
NSZK	049	GR
Olaszország	039	IT
Izrael	972	*
Közel Kelet	785	*
Hollandia	031	*
Norvégia	047	*
Portugália	351	*
Spanyolország	034	*
Svédország	046	*
Svájc	041	*
Egyesült Királyság	044	UK
Egyesült Államok	001	US

A *-gal jelölt országokban nincs külön billentyűzet rutin. Itt tetszőleges yy megadható és egy KEYByy.COM nevű file-nak kell tartalmaznia a billentyűzet rutint.

Megjegyzés:

1. A SELECT parancs a DOS 3.00-nál korábbi verzióiban nem használható.
2. A SELECT parancs végrehajt egy FORMAT parancsot is, tehát a cél-lemezen mindent töröl!
3. A SELECT a cél-lemezen létrehoz egy CONFIG.SYS file-t egyetlen COUNTRY = xxx paranccsal. Ezenkívül a SELECT generál még egy AUTOEXEC.BAT file-t is az alábbi utasításokkal:

```
PATH ;[,út;]
KEYB: yy xxx
DATE
TIME
VER.
```

4.8 SYS - rendszer-file-ok másolása

A parancs lehetővé teszi, hogy az operációs rendszer a megadott meghajtóra kerüljön.

Verzió: 2.11 - 3.30
Parancs típusa: Külső
Parancs formája:

SYS *d*:

Megjegyzés:

1. Az átmásolás a következő esetekben lehetséges:
   - a lemez formázott, de a tartalomjegyzéke tejlesen üres, vagy
   - a lemezt a /B vagy a /S paraméter felhasználásával már formáztuk.
2. Ahhoz, hogy a lemezt rendszerlemezként használjuk, a következőknek kell teljesülni:
   - Az IBMBIO.COM és az IBMDOS.COM file-oknak a tartalomjegyzékben az első két bejegyzésnek kell lenni.
   - Az IBMBIO.COM-nak a legelső két egymást követő adatszektorban kell elhelyezkednie.
3. A PROPOS-16 operációs rendszerben ezt a parancsot nem valósították meg.
4. Néhány alkalmazói rendszert szállítanak úgy, hogy a lemezen már kijelölték a rendszer file-jait, de azok még nem kerültek fel a lemezre. A felhasználó az általa használt DOS verziót a SYS paranccsal helyezheti el a lemezen, amit ezután már rendszerlemezként is használhat.
5. A SYS parancs kiadása után még át kell másolni a cél-lemezre a COMMAND.COM-ot is.

4.9 DIR - tartalomjegyzék kiíratása

A parancs listázza vagy a lemezen lévő összes file-ra, vagy megadott file-ra, illetve file-okra vonatkozó tartalomjegyzék bejegyzéseket. A parancs a következő információkat minden esetben szolgáltatja:

- a lemez azonosítóját (legfeljebb 11 karakter),
- a rendelkezésre álló szabad területet,
- a file-ra vonatkozó információkat:
  - azonosítóját,
  - méretét (decimális számként, byte-okban),
  - az utolsó módosítás dátumát, időpontját.

Verzió: 2.11 - 3.30
A parancs típusa: Belső
A parancs formája:

DIR [*d*:][*út*][*file-név*[.*kit*]][/P][/W]

A paraméterek jelentése:

/P paraméter biztosítja, hogy az információ 24 soronként íródik ki a képernyőre. Egy tetszőleges billentyű lenyomásával folytatódik a kijelzés.

/W paraméter egy rövidített, tömör kijelzést tesz lehetővé. A file azonosítója kerül csak kijelzésre. A 80 oszlopos képernyőre történő kiíratásnál egy sorba öt file-azonosító kerül.

Megjegyzés:

1. A rejtett (hidden) file-oknál nem jelenik meg információ.
2. A tartalomjegyzékben azok a bejegyzések, amelyek altartalomjegyzékre vonatkoznak, < DIR > megjelöléssel kerülnek kijelzésre.
3. A file-ok megadásánál a globális név-jelek is használhatók: a csillag (*) és a kérdőjel (?).
4. A dátum és idő formája a COUNTRY-ban megadott értéknek megfelelően jelenik meg.

A parancs két formában alkalmazható:

I. Alkalmazás - a megadott tartalomjegyzék valamennyi file-jának kijelzése

A parancs formája:

DIR [d:][út]

Ha sem meghajtót, sem útkijelölést nem adunk meg, akkor az aktuális meghajtón lévő lemez aktuális tartalomjegyzékét írja ki.

Az út kijelölése lehetővé teszi egy altartalomjegyzék kiíratását is.

II. Alkalmazás - egy tartalomjegyzék meghatározott elemeiről kapunk információt

A parancs formája:

DIR [d:][file-név[.kit]]

Ha csak a file-nevet adjuk meg, akkor a file-név kiterjesztésénél a csillag (*) a feltételezett. Ha olyan file-t akarunk kiíratni, amelynek nincs kiterjesztése, a neve után a pontot (.) meg kell adni.

Megjegyzés:

1. A tartalomjegyzék kijelzésénél két speciális bejegyzés is előfordulhat. Az egy pont (.) az éppen kijelzés alatt álló altartalomjegyzékre vonatkozik. A két pont mindig az aktuális tartalomjegyzék feletti tartalomjegyzéket jelöli. Így például megengedett a következő parancs:

DIR ..

A tartalomjegyzék kijelzésénél a \ (backslash) a főtartalomjegyzéket jelöli.

Példa:

Amennyiben az altartalomjegyzékeket úgy különböztetjük meg a normál file-októl, hogy nem adunk kiterjesztést, akkor a

DIR *.

parancs ezeket fogja kiírni (azaz pontosan azokat a file-okat, melyeknek nincs kiterjesztése).

4.10 VOL - kötetazonosító kiíratása

A parancs lehetővé teszi, hogy a megadott meghajtón lévő lemez kötetazonosítóját kiírassuk.

Verzió: 2.11 - 3.30
A parancs típusa: Belső
A parancs formája:

VOL [*d*:]

Megjegyzés:

1. Ha meghajtót nem adunk meg, akkor az aktuális meghajtón lévő lemez azonosítóját jelzi ki.
2. Kötetazonosítót a lemeznek formázáskor vagy LABEL paranccsal adhatunk. A kötetazonosító maximálisan 11 karakter hosszúságú lehet.

3. Ha nincs kötetazonosítója a lemeznek, akkor a következő üzenetet kaphatjuk:

   Volume in drive A has no label
   (Az A: meghajtón lévő lemeznek nincs azonosítója.)

4. A DIR parancs az első kiírt sorában szintén kiírja a kötetazonosítót.

4.11 LABEL - kötetazonosító megadása

A parancs segítségével módosíthatjuk egy lemez kötetazonosítóját.

Verzió: 3.00 - 3.30
A parancs típusa: Külső
A parancs formája:

LABEL [*d*:][*cimke*]

Megjegyzés:

1. A *cimke* 1-11 karakterből állhat.

2. Ha nem adjuk meg a cimké-t a parancsban, akkor a DOS kiírja a jelenlegi kötetazonosítót és utána megkérdezi az új kötetazonosítót. Ha erre az ENTER azonnali megnyomásával válaszolunk, akkor újabb kérdés következik:

   Delete current volume label (Y/N)?

(Töröljem a kötetazonosítót (I/N)?) Az Y válaszra a kötetazonosító törlődik, az N-re megmarad az eredeti.

3. A LABEL-t nem szabad a SUBST-tal és az ASSIGN-nal együtt használni.

4. A LABEL hálózaton implementált lemezegységre nem használható.

5. A LABEL a DOS 3.00-nál régebbi verzióiban nem használható.

4.12 DISKCOPY - lemez másolása

A parancs lehetővé teszi, hogy egy floppy lemez tartalmát átmásoljuk egy másik floppy lemezre.

Verzió: 2.11 - 3.30
A parancs típusa: Külső
A parancs formája:

DISKCOPY [*d*: [*d*:]][/1]

A paraméterek jelentése:

Az első meghajtó a forrás-, a második a célmeghajtót jelöli. A /1 paraméter lehetővé teszi, hogy a lemeznek csak az egyik oldalát másoljuk át.

Megjegyzés:

1. A parancs egy-meghajtós rendszerben is használható. Ekkor a program üzenetet küld, és így kéri a felhasználót a lemez cseréjére. Egy billentyű lenyomása után folytatja a működését.

2. Ha számítógépünk lemezegysége egyoldalas lemezhez való, akkor egyoldalas lemezeket másolhatunk egyoldalasokra. Ha kétoldalashoz való, másolhatunk egyoldalast egyoldalasra, és kétoldalast kétoldalasra.

3. A paranccsal több lemez is átmásolható. A másolás befejezése után a program megkérdezi, kér-e a felhasználó további másolást.
   Ha a válasz Y (Igen), akkor a meghajtó kijelölések továbbra is érvényben maradnak. Ha a válasz N (Nem), akkor a parancs befejezi működését.

4. Ha a cél-lemez még nem formázott, vagy nem egyezik a típusa a forrás lemezével, akkor a parancs a lemez formázását is elvégzi.

5. A parancs szektoronként másolja át az információt. A file-ok a céllemezen ugyanabba a szektorba kerülnek, ahol a forrás-lemezen voltak. A lemez átmásolása előtt célszerű a lemez sűrítését is elvégezni. Ez a

   COPY d:*.*

   segítségével történhet, csak ezután célszerű a lemez másolását végrehajtani. A másolás és a sűrítés egy lépésben is végrehajtható, ha a következő parancsot használjuk:

   COPY [d:]*.* d:

6. Ha valamelyik lemezen hiba fordul elő, akkor a DISKCOPY üzenetet küld. Ebben leírja a hiba okát és kijelzi a hibás területet: meghajtó, oldal és sáv. Ezután a program folytatja a másolást. A cél-lemez használhatósága attól függ, hogy a hiba file-t tartalmazó adatterületen fordult-e elő.

7. Ha a DISKCOPY befejezte működését, a két lemez összehasonlítását is célszerű elvégezni. Ezt a DISKCOMP paranccsal végezhetjük el.

8. A DISKCOPY nem használható az ASSIGN-nal és a SUBST-tal együtt.

9. A DISKCOPY nem használható hálózaton implementált egységre. Ilyenkor célszerű helyette a COPY *.* használata.

10. A PROPOS-16-ban járulékos [/C[0]] paraméterek is megadhatók. C paraméter esetén a másolás után ellenőrzés történik, hiba esetén hibajelzéssel. Ha az 0 paramétert is megadjuk, csak összehasonlítás történik, másolás nem.

Példa:

Ha az A: meghajtón lévő lemezt a B: meghajtóban lévő lemezre szeretnénk másolni, akkor a következőképpen célszerű elvégezni:

1. A forrás lemez file-jainak a tömörítése (átmásolással lehetséges):

   C>COPY A:*.* B: /V

2. A lemez visszamásolása:

C > DISKCOPY B: A:

3. A lemezek összehasonlítása:

C > DISKCOMP A: B:

4.13 DISKCOMP - lemezek összehasonlítása

A parancs lehetővé teszi, hogy a DISKCOPY paranccsal történő lemezmásolás után a cél- és a forrás-lemez tartalmát összehasonlítsuk. A paranccsal teljes lemezek összehasonlítása végezhető, míg a COMP csak file-okat hasonlít össze.

Verzió: 2.11 - 3.30
A parancs típusa: Külső
A parancs formája:

DISKCOMP [*d*:[*d*:]][**/1**][**/8**]

A paraméterek jelentése:

A *d*-k a floppy meghajtók azonosítói

/1 paraméter jelöli, hogy csak egyik oldalát kell a lemezeknek összehasonlítani, akkor is, ha azok kétoldalasak.

/8 paraméter esetén a lemezeknek sávonként csak nyolc szektora kerül összehasonlításra, akkor is, ha azok 9 szektor/sáv típusúak.

Megjegyzés:

1. A parancs fix lemez esetén nem használható. Ha a fix lemez azonosítóját adjuk meg, hibaüzenetet kapunk.

2. A parancs egy-meghajtós rendszerben is használható. Ekkor a program üzenettel kéri fel a felhasználót a lemezek cseréjére. Egy tetszőleges billentyű lenyomása után folytatja az ellenőrzést.

3. Ha a DISKCOMP befejezte a két lemez összehasonlítását, akkor üzenetet küld. Megkérdi a felhasználót, hogy akarja-e még az összehasonlítást folytatni. Ha igen, akkor a válasz Y. Az összehasonlítás ugyanazokon a meghajtókon történik. Ha nem, akkor a válasz N.

4. A program sávonként végzi az összehasonlítást. Üzenetet küld, ha ezek nem egyeznek. Ebben megadja az eltérő sáv sorszámát (0-39.), és azt, hogy melyik oldalon található (0-1.).

5. Ha nem adunk meg meghajtó azonosítót, akkor az összehasonlítás az aktuális meghajtón történik, mintha csak egy meghajtó lenne a rendszerben.

6. Ha nem adunk meg paramétert, akkor a lemez típusának megfelelően végzi az összehasonlítást:

   - Mindkét oldal dupla oldalú lemezként.
   - 9 szektor/sáv.

7. Ha a COPY paranccsal másoltunk egy lemezt, akkor a DISKCOMP paranccsal történő összehasonlítás hibát jelez, mivel a COPY a file-okat általában nem ugyanarra a helyre másolja a cél-lemezen, mint ahol az a forrás lemezen volt.

   A DISKCOMP az összehasonlítást oldalanként, sávonként, szektoronként, byte-ról byte-ra végzi.

8. A parancs a hiba kijelzése után mindig folytatja az összehasonlítást, függetlenül attól, hogy eddig hány eltérést jelzett. (A COMP tíz eltérés után befejezi a működését.)

9. A DISKCOMP parancs nem használható a SUBST vagy a JOIN paranccsal együtt.

10. A DISKCOMP nem tudja értelmezni az ASSIGN-ban megadott értékeket.

11. A DISKCOMP hálózaton implementált lemezegységre nem használható.

Példa:

1. Ha csak a következőt adjuk meg:

   A > DISKCOMP

akkor az A: meghajtón történik az összehasonlítás. A meghajtóba először a forrás lemezt kell betenni, majd a rendszer üzenetet küld és felkér a lemezek cseréjére. Ha kész, le kell nyomni egy billentyüt, és az öszehasonlítást folytatja.

Ha befejezte az összehasonlítást, akkor megkérdezi, kell-e további összehasonlítást végeznie. Az Y, illetve a J válasz esetén folytatja az összehasonlítást, N válasz esetén a program befejezi működését.

4.14 CHKDSK - lemez vizsgálata

A parancs információt ad a lemez állapotáról a következő paraméterekkel:

- kötetazonosító,
- a lemez teljes kapacitása,
- védett file-ok mérete,
- felhasználói file-ok mérete,
- a szabad lemezterület nagysága,
- a RÀM memória mérete,
- a szabad RAM memória mérete.

Helyreállítja a file-okat, ha ellentmondás van a file leíró táblázat (file allocation table - FAT) és a tárolt adatterületek között. A parancs hálózaton implementált lemezegységre nem adható ki!

Verzió: 2.11- 3.30
A parancs típusa: Külső
A parancs formája:

CHKDSK [*d*:][*file-név*[*.kit*]][/F][/V]

A paraméterek jelentése:

Ha egy file nevét adjuk meg, akkor a file nem összefüggő területeinek a számát adja meg.

/F paraméter megadásakor a program felújítja a FAT-ot. A FAT javítását a fizikai tárolásnak megfelelően végzi el. Ha nem adjuk meg a /F paramétert, akkor hibás FAT esetén csak rövid hibaüzenetet küld és kijelzi a lehetséges javítást, de azt nem végzi el.

/V paraméter megadása esetén a CHKDSK részletes üzenetet küld.

Megjegyzés:

1. A CHKDSK csak a fő tartalomjegyzéket használja.
2. Ellenőrzés után a parancs a lemez állapotáról információkat is küld.
3. A program a védett file-okról is ad információt.
4. A CHKDSK nem vár arra, hogy feltegyük a lemezt, ezért egy cserélhető lemezegységet tartalmazó rendszer esetén meg kell adni a kívánt meghajtót. Ha nem adjuk meg, akkor az aktuális meghajtón lévő lemezt ellenőrzi.
5. A parancs fix lemez és floppy esetében is használható.
6. Minden javítást kijelez a CHKDSK, az Y és az N beadása után az üzenet végét jelző billentyűt (< ENTER >) is le kell nyomni.
7. A globális név-jelek használhatók. Ha a következő parancsot adjuk meg:

   CHKDSK [d:]*.*

   akkor a lemezen a file-ok tördeltségéről kapunk információt.
8. Ha a CHKDSK különálló adatterületeket talál, akkor rákérdez, hogy a file-ok különálló területeit összegyűtse-e.

   Ha a /F paramétert megadjuk és a válaszunk Y, akkor a területeket a következő azonosítóval gyűjti össze:

   FILEnnnn.CHK

   ahol nnnn a file sorszáma.

   A file-ok azonosítóját a tartalomjegyzékbe írja. A felhasználó ezután ellenőrizheti, hogy számára hasznos információt tartalmaznak-e. Ha nem, akkor törölheti ezeket.
9. A PROPOS-16 rendszerben a CHKDSK funkcióját a STAT parancs látja el.

4.15 VERIFY - írás utáni ellenőrzés

A parancs a lemezre történő írás ellenőrzését biztosítja.

Verzió: 2.11 - 3.30
A parancs típusa: Belső
A parancs formája:

VERIFY [ON | OFF]

A parancs működése:

A VERIFY ON kiadása után a rendszer minden lemezes írásművelet után ellenőrzi, hogy az éppen felírt információ visszaolvasható-e. Ez jelentősen megnöveli az I/O műveleti időket. Használatát csak indokolt esetben ajánljuk. A VERIFY OFF kiadásával az írás-ellenőrzést kikapcsoljuk. Ha csak a VERIFY kulcsszót adjuk be, akkor a kapcsoló aktuális értékét jelzi ki a DOS (ON-OFF).

Megjegyzés:

Néhány parancs működése alatt is megadható az írásellenőrzés lehetősége. Például a COPY /V paramétere.

Példa:

Ha a VERIFY kapcsoló értékét akarjuk lekérdezni, akkor az a következőképpen történhet:

VERIFY

VERIFY =OFF, ha a kapcsoló nem volt, és VERIFY =ON ha a kapcsoló be volt kapcsolva.

4.16 SUBST - tartalomjegyzék helyettesítése lemezegység betűjelével

A parancs lehetőséget biztosít arra, hogy egy adott tartalomjegyzékre, mint új lemezegységre hivatkozhassunk. Ez főleg olyan felhasználói programok esetén jó, amelyik nem ismeri a tartalomjegyzékeket.

Verzió: 3.10 - 3.30
Parancs típusa: Külső
Parancs formája:

SUBST [*d: d1:út*] | [*d:* **/D**]

A továbbiakban a *d:* lemezegység a *d1:út* által kijelölt tartalomjegyzékkel lesz azonos. A *d:*/**D** alak használata megszünteti a kijelölés hatását.

Példa:

SUBST E: C:\WORK

A fenti parancs kiadása után az E: lemezegység logikailag azonos lesz a C:\WORK tartalomjegyzékkel. A SUBST E:/D visszaállítja az eredeti állapotot.

Megjegyzés:

1. Ha a SUBST parancsot többször is használjuk, akkor a CONFIG.SYS file-ban el kell helyezni egy megfelelő LASTDRIVE = parancsot is.
2. A helyettesített lemezre ne használjuk a CD, MD, RD és PATH parancsokat.
3. SUBST használatakor nem használhatjuk az alábbi parancsokat: ASSIGN, BACKUP, DISKCOMP, DISKCOPY, FDISK, FORMAT, JOIN, LABEL, RESTORE.
4. A paraméter nélküli SUBST kiírja az érvényben levő helyettesítéseket. A fenti példa esetén:

   E: = > C:\WORK

4.17 JOIN - lemezegység és tartalomjegyzék összekapcsolása

A parancs segítségével egy lemezegységet egy tartalomjegyzékkel logikailag összekapcsolhatunk.

Verzió: 3.10 - 3.30
A parancs típusa: Külső
A parancs formája:

JOIN [*d1: d2:\tartalomjegyzék*] | [*d:*/**D**]

A paraméterek jelentése:

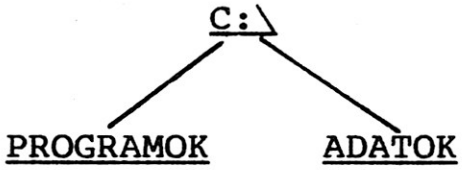
A *d1:* lemezegységet kapcsolja össze (azonosítja) a JOIN parancs a *d2:* lemezegység *tartalomjegyzékével*. Mint a parancs alakjából is látható, ez a tartalomjegyzék közvetlenül a főtartalomjegyzékben kell, hogy elhelyezkedjen.

A *tartalomjegyzék* a *d2:* egységen vagy üres kell, hogy legyen, vagy egyáltalán nem létezik.

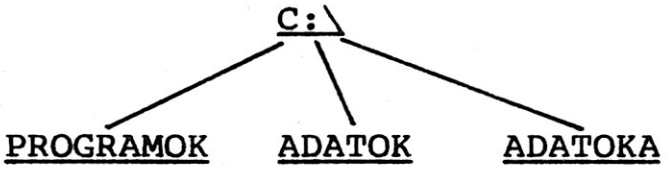
A /**D** paraméter törli egy előzőleg létrehozott JOIN hatását.

Megjegyzés:

1. A JOIN segítségével az adott lemezhez egy üres altartalomjegyzéket kapcsolunk hozzá, amelyik a másik lemezegységet reprezentálja. Tegyük fel, hogy a C: lemezegység altartalomjegyzékei a következőképpen helyezkednek el:



A JOIN A: C:\ADATOKA parancs kiadása után az a struktúra így alakul:



2. Az aktuális lemezegység nem kapcsolható össze egyetlen altartalomjegyzékkel sem. Ha megkíséreljük, akkor INVALID PARAMETER hibaüzenetet kapunk.

3. Kapcsolt lemezegységre nem lehet hivatkozni. Ha az 1. megjegyzésben szereplő szituációban kiadjuk a DIR A: parancsot, akkor INVALID DRIVE SPECIFICATION hibaüzenetet kapunk. Helyette a CD ADATOKA, majd a DIR parancsot kell kiadni.

4. A JOIN hálózaton implementált lemezegységre nem használható.

5. A SUBST, illetve ASSIGN használatával együtt a JOIN előre nem látható hibákat eredményezhet.

6. A BACKUP, RESTORE, FORMAT, DISKCOPY és a DISKCOMP parancsokat ne használjuk addig, míg a JOIN érvényben van!

7. A paraméter nélküli JOIN kiírja az adott összekapcsolásokat. Az 1-beli szituációban a JOIN kiadása után az alábbi üzenetet kapjuk:

   A: = > C:\ADATOKA

8. A JOIN a DOS 3.1-nél régebbi változataiban nem használható.

4.18 APPEND - bejárási utak kijelölése tetszőleges file-hoz

Az APPEND parancs lehetővé teszi olyan file-ok elérését is, amelyek az aktuális tartalomjegyzéken kívül vannak, anélkül, hogy az eljárási utat mindig újból-és újból leírnánk. Egyidőben több elérési út is kijelölhető. A paranccsal egyben le is kérdezhetjük a kijelölt utakat.

Verzió:3.30
A parancs típusa: Külső, majd rezidens
A parancs formája:

APPEND *d:út*[;[*d:*]*út*...]
APPEND [/X][/E]
APPEND [;]

A paraméterek jelentése:

d:út annak az elérési útnak amegadása, amit a DOS-nak - az aktuális mellett - végig kell néznie.

/X a kapcsoló utasítja a programot arra, hogy a DOS 11H, 4EH és 4BH funkcióinál megadott file-okat is keresse a benne megadott utakon. Különben ezt csak a 0FH, 23H és 3DH funkciókra vonatkozik.

/E a kapcsoló megadásakor az APPEND-ben megadott utakat a programkörnyezetben (environment) helyezi el a parancs.

Megjegyzések:

1. Az APPEND első hívásakor csak az 1. és 2. formát használhatjuk. Ilyenkor a parancs előtt úttal is megadhatjuk, hogy melyik tartalomjegyzékben található. Ezt követően az APPEND a memóriában marad, a DOS rezidens részévé válik. Ezért az APPEND előtt már nem szerepelhet útkijelölés.

2. Ha már egyszer a memóriába töltöttük az APPEND programot, akkor a parancs 1. és 3. formáját használhatjuk.

3. Mindig az utoljára használt APPEND-ben levő út-megadások érvényesek. Ezért az APPEND; törli az összes utat.

4. A paraméter nélküli APPEND a képernyőre írja az éppen érvényes bejárási utakat.

5. Hálózaton installált egységen az APPEND nem használható.

6. Az APPEND segítségével használt utakon csak olvasni lehet!

7. Szemben a PATH-tal, amely a bejárási utak mentén levő file-ok közül csak a .COM, .EXE és .BAT kiterjesztésűeket találja meg, az APPEND-del tetszőleges kiterjesztésű file elérhető.

5. DOS: a fix lemez kezelése

A PC DOS, néhány korai verziójától eltekintve, támogatja a fix lemez kezelését. Ha a fix lemezre írni vagy arról olvasni szeretnénk, akkor a meghajtó azonosítóját kell megadni. Ez általában az

- első fix lemez esetén a C,
- a második fix lemez esetén D.

A fix lemez négy különböző méretű területre osztható fel. A területeket partíciónak nevezzük. A partíciók mindegyikét egy-egy operációs rendszer használhatja. Lehetőségünk van viszont arra is, hogy az egész területet egy operációs rendszer kezelje. A különböző operációs rendszerek mindegyike tartalmaz olyan komponenst, amely a partíciók kezelését végzi. A PC DOS rendszerben ez az FDISK program.

A DOS 3.20-as és korábbi változatai nem tették lehetővé, hogy a DOS több partíciót is használjon. Erre nem is volt eleinte szükség, hiszen a lemezegységek mérete kicsi volt. A lemezegységek kapacitásának növelése azonban megkívánta, hogy több partíciót is használhasson a DOS . A partíciók szerkezete ugyanis olyan, hogy maximum 32 Mbyte információ fér rá. Így az ennél nagyobb egységek 'felesleges' kapacitását a DOS maga már nem tudta kezelni.

A DOS 3.30-as verziója kétféle DOS partíciót ismer. Az első az ún. elsődleges DOS partíció, a másik az ún. kiegészítő DOS partíció. Angol elnevezésük: primary, ill. extended. A fixlemezen egy elsődleges és egy kiegészítő DOS partíció rendelhető. A kiegészítő DOS partíció területét ezután logikai egységekre oszthatjuk. Egy logikai egység maximum 32 Mbyte lehet. Így tetszőleges méretű fix egységet képes a DOS kezelni, azzal a megkötéssel, hogy a kiegészítő DOS partícióban levő logikai egységek maximum 32 Mbyte naagyságúak lehetnek.

5.1 FDISK - fix lemez kialakítása

Az FDISK program lehetővé teszi, hogy a fix lemezen

- DOS-partíciót előállítsunk,
- DOS-partíciót aktivizáljunk,
- DOS-partíciót töröljünk,
- a DOS-partíció jellemzőit kiírassuk,
- áttérjünk a másik fix lemezre.

Verzió: 2.11 - 3.30
A program azonosítója: FDISK.COM

Indítása tehát úgy történik, hogy az FDISK kulcsszót beírjuk. Az indításkor a programnak nem kell paramétereket megadni. A program a képernyőre kiírja a választható lehetőségeket (a menüt). A megfelelő lehetőség kiválasztása a mellé írt szám megadásával történik.

Az FDISK programhoz az ESC billentyű lenyomásával térhetünk vissza, ismételt lenyomásával pedig a DOS rendszerhez.

A fix lemez szerkezete

A fix lemez cilinderekre van felosztva. A cilinderek száma és mérete a fix lemez típusától függ. A 10 Mb-os lemez 305 cilindert tartalmaz.(A 20 Mbyte-os általában 611 cilindert.) Egy cilinder kapacitása 32768 byte.

Az első cilinder sorszáma: 000, az utolsóé pedig: 304.

Az aktív partíció azt a DOS-partíciót jelöli, amelyről a DOS rendszer betölthető. Abban az esetben, ha két fix lemez van a rendszerben, csak az elsőn lehet aktív partíciót kijelölni. A DOS rendszert csak az első fix lemezről lehet betölteni.

A következőkben ismertetjük a különböző választási lehetőségek lépéseit. A menü pontjai és azok jelentése a következő:

1. Create DOS Partition	DOS partíció létrehozása
2. Change Active Partition	az aktív partíció kijelölése
3. Delete DOS Partition	DOS partíció törlése
4. Display Partition Data	a partícióadatok kiíratása
5. Select Next Fixed Disk Drive	a következő fix lemezegység kiválasztása

5.1.1. DOS-partíciók előállítása

Elsődleges DOS-partíció:

1. lépés: Az FDISK indítása. Ezután a képernyőn megjelenik a főmenü.

2. lépés: Az 1 választásával és beírásával, majd az < ENTER > megnyomásával a DOS-partíció előállítását választjuk ki.

3. lépés: A program megkérdezi, hogy elsődleges vagy kiegészítő DOS partíciót kívánunk-e létrehozni. Most az 1-es pontot kell választanunk. A DOS 3.20-as vagy korábbi verzióiban ez a kérdés kimarad, azokugyanis nem képesek csak az elsődleges DOS partíció kialakítására.

Ekkor a DOS újabb üzenetet küld, amelyben arról kér információt, hogy a teljes fix lemezt a DOS számára tartjuk-e fenn, vagy több partícióra (legfeljebb 4) szeretnénk osztani. (Megkérdezi: "Do you wish to use the entire fixed disk for DOS (Y/N)" - kívánjuk-e a teljes lemezt fenntartani a DOS számára - N-t válaszolunk, ha több partíciót akarunk.) Ha az egész fix lemezt a DOS alatt akarjuk használni, akkor csak az üzenet végét jelző billentyűt (< ENTER >) kell lenyomni.

4. lépés: A DOS újraindítása az A: vagy a B: meghajtóról egy tetszőleges billentyű lenyomásával történik. Ha a DOS-partíciót a második fix lemezen hoztuk létre, akkor a rendszert nem kell újraindítani.

5. lépés: A fix lemez DOS partíciójának előkészítése a file-ok befogadására a

FORMAT C:/S/V

vagy

FORMAT C:/V

paranccsal történik.

A C: a fix lemez azonosítója. A /S paraméter hatására a rendszer-file-okat is átmásolja, a /V paraméter megadása esetén pedig legfeljebb 11 jelből álló kötetazonosítót helyez el a lemezen a rendszer.

Ha az egész DOS rendszert el akarjuk helyezni a lemezen, akkor a parancs a következő:

COPY *.* C:

Az aktuális meghajtóról valamennyi file átkerül a fix lemezre.

6. lépés: A DOS rendszer indítása fix lemezről. Az A: meghajtóból kivesszük az eredeti DOS lemezt. A <Ctrl-Alt-Del> együttes lenyomásával már a fix lemezről indíthatjuk újra a rendszert.

Megjegyzés:

Ebben az esetben az A: meghajtóban nem lehet DOS rendszert tartalmazó lemez. A számítógép először az A: meghajtóról próbál rendszert tölteni. Ha ez eredménytelen, csak ezután fordul az első fix lemezhez.

A dátum és az idő megadása után már új egységazonosító jelenik meg (C>), amely az első fix lemezt azonosítja. A DOS aktuális meghajtónak most a C: meghajtót tekinti.

Ha már a fix lemezt más rendszerek alatt használtuk, akkor informatív üzenet jelenik meg a partíciókról. Ez tartalmazza a partíció

- azonosítóját (1.-4.),
- állapotát (A-aktív, N-nem aktív),
- típusát (DOS partíció-e),
- kezdetét (kezdő cilinder),
- végét (utolsó cilinder),

valamint a teljes lemezterület méretét cilinderben, a még rendelkezésre álló szabad terület méretét és kezdő címét. Abban az esetben, ha két fix lemez van a rendszerben, a meghajtó azonosítóját is kiírja.

Ha a lemezen még nincs 4 partíció kijelölve és szabad terület is van, akkor ezen a területen létrehozhatunk egy DOS-partíciót.

Kiegészítő DOS partíció (csak 3.30-as verziótól)

Ha a lemezen nincs még négy partíció kijelölve, és a meglevő partíciók egyike sem kiegészítő partíció, akkor lehetőségünk van annak megadásaira. A művelet egyes lépései a következők:

1. lépés: Az FDISK indítása. Ezután a képernyőn megjelenik a főmenü.

2. lépés: Az 1 beírásával és az <ENTER> megnyomásával a DOS partíció előállítását választjuk.

3. lépés: A program megkérdezi, hogy elsődleges vagy kiegészítő DOS partíciót kívánunk-e létrehozni.Most a 2-es pontot választjuk, a kiegészítő DOS partíció megadását.

Ekkor a DOS újabb üzenetet küld, melyben jelzi, hogy még hány cilinder áll rendelkezésre. A program válaszul a maximálisan rendelkezésre álló cilinderek számát kínálja. Egyébként tetszőleges - ennél kisebb - értéket adhatunk meg.

Újabb üzenet közli, hogy a kiegészítő DOS partíciót létrehozta a rendszer.

4. lépés: Logikai egységek hozzárendelése a kiegészítő DOS partícióhoz. Az <ESC> megnyomása után újabb üzenetet kapunk, hogy még nincs logikai egység definiálva. A program az előző logikai egység nagyságát kéri, s válaszul a teljes kiegészítő DOS partíciót kínálja. Tetszőleges számú és nagyságú logikai egységet létrehozhatunk, ezek összmérete meg kell, hogy egyezzen a kiegészítő DOS partíció méretével.

5.1.2. Aktív partíció változtatása

Ezt a lehetőséget akkor választjuk, ha az operációs rendszert egy másik partícióból szeretnénk indítani.

1. lépés: Az FDISK indítása. Ezután a képernyőn megjelenik a választható lehetőségek listája.

2. lépés: A 2 megadásával az aktív partíció változtatását választjuk ki. Ezután a DOS informatív üzenetet küld a partíciókról, és bekéri az aktivizálandó partíció azonosítóját. Az <ESC> billentyű lenyomásával az FDISK főmenühöz térhetünk vissza. Az <ESC> ismételt lenyomásával pedig a DOS-hoz.

A számítógép továbbra is az eredeti partícióval dolgozik. Ha az új aktív partícióról akarjuk a rendszert indítani,akkor a <Ctrl-Alt-Del> billentyűkombinációval újra kell tölteni a fix lemezről a rendszert. Ilyenkor az A: meghajtóban természetesen nem lehet lemez.

Megjegyzés: Aktív partíció csak az első fix lemezen lehet.

5.1.3. DOS-partíció törlése

Ez a file-ok törlését is jelenti. Ezért, ha ezt a lehetőséget akarjuk választani, előtte a file-okat ajánlatos kimenteni. Az aktív DOS-partíciót nem lehet törölni, előtte át kell állítani (2. választási lehetőség).

A végrehajtás lépései hasonlítanak az aktív partíció változtatásánál megismertekhez, de értelemszerűen a 2. lépésben a 3 megadásával választható ki a DOS-partíció törlése funkció.

5.1.4. A partícióadatok kiíratása

Ezt a lehetőséget akkor választjuk, ha a fix lemez felosztásáról szeretnénk információt kapni.

A képernyőn a következő információk jelennek meg:

- a partíció azonosítója,
- állapota (aktív - nem aktív),
- típusa,
- kezdete,
- vége,
- mérete.

Megkapjuk a teljes lemezterület méretét (cilinderben). A fix lemez meghajtója csak abban az esetben kerül kijelzésre, ha a rendszerben több fix lemez van.

Ha a DOS 3.30-as vagy későbbi verzióit használjuk, akkor a program megkerdezi, hogy a kiterjesztett DOS partícióhoz rendelt logikai egységek adatait is kijelezze-e. Ha <4>-et, majd <ENTER> nyomunk le, akkor a képernyőre kerülnek ezek az adatok is.

Az üzenet utolsó sorában az FDISK főmenühöz való visszatérés módja van leírva: <ESC> billentyű lenyomása.

A kiírás nem tartalmaz információt a fix lemez szabad területéről.

5.1.5. A következő fix lemez kiválasztása

Ezt a lehetőséget csak akkor választhatjuk, ha a rendszerben legalább két fix lemez van. Ha ezt választottuk, akkor az FDISK "átlép" a másik fix lemezre. A képernyőn megjelenő FDISK főmenüben a pillanatnyi fix lemez meghajtó azonosítója megváltozik. Például, ha kezdetben 1 volt, az "átlépés" után 2 lesz. Ez lehetővé teszi, hogy az FDISK ismételt behívása nélkül két fix lemez felosztását is elvégezzük. A DOS rendszerhez az <ESC> billentyű lenyomásával térhetünk vissza.

5.2 Fix lemez archiválása

A fix lemez teljes, vagy meghatározott file-jainak kimentését floppy-ra, és arról visszatöltését a BACKUP és a RESTORE parancs segítségével lehet elvégezni. A file-ok speciális formában kerülnek a floppy-ra, így ezek nem használhatók fel, csak a fix lemezre való visszatöltés után. Ha a fix lemez tartalma nem fér el egy floppy lemezen, a DOS folytatás kötetet kér. Természetesen ezeket a visszatöltés során is tudja kezelni.

5.2.1 BACKUP - file-ok archíválása

A fix vagy floppy lemez teljes tartalmát, vagy meghatározott file-jait menti floppy-ra vagy fix lemezre.

Verzió: 2.11 - 3.30
A parancs típusa: Külső
A parancs formája:

BACKUP *d:*[*út*][*file-név*[*.kit*]] *d:*[**/S**][**/M**][**/A**][**/D:***hh-nn-éé*]
[**/T:***óó:pp:mm*][**/F**][**/L**[**:**[*d:*][*út*]*file-név*[*.kit*]]]

A paraméterek jelentése:

Az első paramétersor a mentésre kijelölt file-okat írja le. A második paramétersor azt a meghajtót azonosítja, amelyre a mentés történik. A floppy lemezeket előzőleg meg kell formázni.

A forrás- és a cél lemezegységnek (d:) különbözőknek kell lennie. Mind floppy egység, mint Winchester tetszőleges kombinációban szerepelhet forrásként és célként.

Azon lemezegységet, amelyik nem az aktuális, meg kell adni! (d:)

/S	megadásával a megadott, illetve ha nem adtuk meg, az aktuális tartalomjegyzékhez tartozó összes file kimentésre kerül (az összes, ezen tartalomjegyzékben levő altartalomjegyzékbeli file-ok is)
/M	megadása esetén csak azok a file-ok kerülnek mentésre, amelyeket az utolsó archíválás óta módosítottunk
/A	megadása esetén a cél-lemezen már meglévő file-ok nem törlődnek, egyébként csak a kimentett file-ok lesznek a cél-lemezen.
/D:*hh-nn-éé*	megadásával egy időpontot jelölünk ki, amelytől kezdve felvitt, illetve módosított file-ok kerülnek lementésre.
/T:*óó:pp:mm*	megadásakor csak azok a file-ok kerülnek mentésre, amelyek az adott naptól és időponttól kezdve valamikor módosításra kerültek, vagy akkor hozták létre.
/L	segítségével a lemezre LOG file-t írhatunk. Ennek a neve BACKUP.LOG lesz. Az első rekord a mentés dátumát és időpontját tartalmazza. A következő rekordok tartalma: a mentéslemez sorszáma, és minden file teljes neve.
/L:[*d:*][*út*]*file név*[*,kit*]]	segítségével módosíthatjuka LOG file nevét.

Megjegyzés:

1. A file létrehozásakor és minden módosításakor a tartalomjegyzékbe bekerül a napi dátum és az idő is. Ezért lényeges, hogy a rendszer indításakor megadjuk a pontos időt és a dátumot.

2. A PROPOS-16 operációs rendszerben /C paraméterrel másolás utáni ellenőrzés kérhető.

3. A BACKUP és a RESTORE parancsok az IBM PC DOS, illetve az MS DOS esetén nem ugyanazok. Az egyes verziók sem kompatibilisek egymással. A fenti formátum az IBM PC DOS BACKUP parancsa, az MS DOS ettől eltér:

BACKUP [*d*:][*út*][*file-név*[*.kit*]][*d*:][/**S**][/**M**][/**A**][/**D**:*hh-nn-éé*] [/**P**][/**T**:*óó:pp:mm*][/**L**[:*file-név*]]

Az új paraméter használata a következő:

/P a megadjuk ezt a paramétert, akkor annyi file-t másol egyetlen hajlékonylemezre, amennyit csak tud. Ehhez esetenként egy altartalomjegyzéket is létrehoz. (Ennek az opciónak a használata nem kompatibilis az IBM PC DOS BACKUP parancsával!)

4. Az /L ésa /T opciók a DOS 3.20-as és korábbi verziókban nem használható! Az MS DOS rendszerben azonban ezek a paraméterek is használhatók.

A BACKUP működése:

Ha nem adjuk meg a /A paramétert, akkor a floppy vagy egyéb cél lemez valamennyi file-ját törli, mielőtt a forrás lemez file-jainak a mentését elkezdené. Ha floppy-ra mentünk, és a lemez megtelt, akkor felkéri a felhasználót, hogy cserélje ki a floppyt. Az archíválásra kerülő file-ok azonosítóját kijelzi.

A BACKUP a következő módon állítja a befejezési kódot:

0 - Normális befejezés.
1 - A file nem található.
2 - File megosztási probléma miatt néhány file nem került elmentésre.
3 - Felhasználó kényszerítette a mentés befejezését (<Ctrl-Break>).
4 - Permanens I/O hiba.

Ezt a kódot a DOS parancsokból képezhető úgynevezett parancs file-okban felhasználhatjuk a parancsok feltételes végrehajtására (lásd 7. fejezet IF ERRORLEVEL parancs utasításforma).

Megjegyzés:

1. A file-ok kijelölésénél a globális név-jelek használhatók.
2. Ha az utat nem definiáljuk, akkor az aktuális tartalomjegyzék file-jai kerülnek mentésre.
3. A BACKUP paranccsal kimentett file-ok normál körülmények között nem használhatók. Felhasználás előtt a RESTORE paranccsal kell visszatölteni ezeket.
4. A parancs minden mentési lemezen létrehoz egy BACKUPID.@@@ file-t, amelyik a lemez sorszámát tartalmazza.

Példa:

1. Ha a következőt adjuk be:

```
BACKUP C:*.TXT A:
```

akkor a C: meghajtó aktuális tartalomjegyzékének valamennyi .TXT kiterjesztésű file-ja az A: meghajtóban levő floppy(k)ra kerül.

2. A következő paranccsal a C: meghajtó valamennyi file-ja mentésre kerül az A: meghajtón lévő lemezre.

BACKUP C:\ A:/S

A \ (backslash) a főtartalomjegyzéket jelöli ki, a /S paraméter pedig az összes altartalomjegyzéket és azok file-jait.

5.2.2 RESTORE - file-ok visszatöltése

Az előzőleg BACKUP paranccsal mentett file-ok visszatöltését végzi. Lehetőség van valamennyi, vagy meghatározott file-ok visszatöltésére.

Verzió: 2.11 - 3.30
A parancs típusa: Külső
A parancs formája:

RESTORE *d:* [*d:*][*út*][*file-név*[.*kit*]][**/S**][**/P**][**/B**:*dátum*][**/A**:*dátum*][**/E**:*hh-nn-éé*] [**/L**:*óó-pp-mm*] [**/M**][**/N**]

A paraméterek jelentése:

Az első paraméter azt a meghajtót jelöli, amelyről a visszatöltés történik. A második paraméter a cél lemez file-jait azonosítja.

/S paraméter megadásával a mentett file-ok valamennyi altartalomjegyzékbe visszatöltésre kerülnek.

/P paraméter megadása esetén a RESTORE üzenetet küld, mielőtt visszatöltené azokat a file-okat, amelyek az utolsó mentés óta nem változtak vagy csak olvasásra hozzáférhetők (Read Only). A két rendszer-file:

IBMBIO.COM és az
IBMDOS.COM

is védett file.

/B dátum megadásával csak a szóbanforgó dátumkor vagy előtte módosított file-okat tölti vissza

/A dátum megadásakor csak a szóbanforgó dátum idején vagy az után módosított file-okat tölti vissza

/E idő megadásakor csak az adott időpontban vagy azt megelőzően módosított file-okat tölti vissza

/L idő megadásakor csak az adott időpillanatban vagy azt követően módosított file-okat tölti vissza

/M esetén csak azokat a file-okat tölti vissza, amelyeket az utolsó elmentés óta módosítottunk

/N esetén csak azokat a file-okat tölti vissza, amelyek nem léteznek a cél lemezen.

Megjegyzés:

1. Csak azok a file-ok tölthetők vissza, amelyeket előtte BACKUP paranccsal mentettünk le.

2. Ha az út paramétert nem adjuk meg, akkor a file-ok az aktuális tartalomjegyzéknek megfelelően kerülnek visszatöltésre.

3. Ha file-azonosítókat nem adunk meg, akkor valamennyi mentett file visszatöltésre kerül.

4. A globális név-jelek használhatók.

5. Ha a mentés több floppy-ra történt, a RESTORE ellenőrzi, hogy a felhasználó a lemezeket megfelelő sorrendben helyezte-e el. Ha nem a megfelelő folytatás kötetet tette fel, akkor üzenetben felkéri a cserére.

6. Vigyázat! Ha a rendszer-file-okat (IBMBIO.COM és IBMDOS.COM) is lementettük BACKUP-pal, ezeket NE töltsük vissza a fix lemezre, mert az nem lesz boot-olható! Ilyenkor a /P paramétert használjuk és a két rendszer-file-nál N-et válaszoljunk az engedélyezést kérő üzenetre.

7. A RESTORE a következő módon állítja a befejezési kódot:

   0 - Normális befejezés.
   1 - A file nem található.
   2 - File megosztási probléma miatt néhány file nem került visszatöltésre.
   3 - Felhasználó kényszerítette a mentés befejezését (<Ctrl-Break> ill. <ESC>).
   4 - Permanens I/O hiba a visszatöltés alatt.

   Ezt a kódot a DOS parancsokból képezhető úgynevezett parancs file-okban felhasználhatjuk a parancsok feltételes végrehajtására (lásd 7. fejezet IF ERRORLEVEL utasításforma).

Példa:

1. A következő parancs megadásával minden .TXT kiterjesztésű mentett file visszatöltésre kerül az A: meghajtóról az első fix lemezre:

```
RESTORE A: C:*.TXT
```

2. Ha valamennyi mentett file-t szeretnénk visszatölteni, akkor a következőt kell megadni:

```
RESTORE A: C:\/S/P,  illetve

RESTORE A: C: /S /P  (ha a rendszerfile-okat is lementettük.)
```

3. Ha egy megadott altartalomjegyzék valamennyi mentett file-ját szeretnénk visszatölteni, akkor a következőt adjuk meg:

```
RESTORE A: C:\ALTART1\*.*
```

6. A rendszer működését befolyásoló parancsok és kapcsolók

Ebben a fejezetben azokat a parancsokat ismertetjük, amelyekkel a rendszer állapotát változtathatjuk meg, illetve amelyek a rendszer állapotáról adnak információt. Így például:

- a képernyő törlése,
- a dátum és az idő megadása,
- a DOS verziójának lekérdezése,
- a < Break > billentyű figyelése,
- periféria-hozzárendelések és működésmódjuk megváltoztatása,
- a klaviatúra kiválasztása,
- grafikus jelek kezelése.

6.1 COMMAND - Újabb parancsfeldolgozó program elindítása

Verzió: 2.11 - 3.30
A parancs típusa: Külső
A parancs formája:

COMMAND [*d*:][*út*][/P][/C *karaktersorozat*][/E:*xxxxx*]

A paraméterek jelentése:

A DOS az adott meghajtón és '*út*'-on keresi a parancsfeldolgozót, és betölti a memóriába. Ha a /P paramétert megadjuk, a felhasználói parancsprocesszor a DOS újraindításáig a memóriában marad. A /C paraméterrel a parancsprocesszornak a megadott *karaktersorozat*ot adhatjuk át, majd az automatikusan visszatér a rendszer eredeti parancsfeldolgozójához. Ha a /P és a /C paramétereket egyszerre adjuk meg, a /P-t a rendszer figyelmen kívül hagyja. /E:*xxxxx* egy decimális szám, amelyik megadja a programkörnyezet (environment) nagyságát. Ez az érték a 160-32768 intervallumba kell hogy essen. Ez a paraméter csak a DOS 3.20 és későbbi verzióiban használható. (Az environmentről lásd 15.3 paragrafust.)

Például:

COMMAND /C DIR C:

Betöltődik a másodlagos parancsprocesszor és a DIR C: parancsot végre is hajtja, majd törlődik a memóriából.

Megjegyzés:

1. A felhasználói parancsprocesszor betöltése és használata után, ha nem adtuk meg sem a /P, sem a /C paramétert, a meghívó programhoz az EXIT paranccsal lehet visszatérni.

2. Ismét felhívjuk a figyelmet, hogy a DOS újraindítása után ismét a standard COMMAND.COM lesz a parancsprocesszor.

3. A DOS ezen tulajdonságát használja fel sok program DOS parancsok kiadására és programok indítására.

   A felhasználói programok is hívhatják az új parancsprocesszort. A felhasználói program paramétereket is átadhat az új parancsprocesszornak, amit az úgy hajt végre, mintha ezt a karaktersorozatot az aktuális input egységről olvasta volna.

A felhasználói parancsprocesszor hívásának lépései:

1. A memóriában 17 Kbyte szabad területet kell biztosítani az új parancsprocesszornak és paramétereinek. Meg kell győződni, hogy van-e még ekkora szabad terület.

2. Az új parancsprocesszornak fel kell építeni a paraméter mezőt. Ennek formája a következő:

1. byte tartalma:	a karaktersorozat hossza,
xx byte tartalma:	a karaktersorozat,
utolsó byte tartalma:	hexadecimális 0D (decimális 13), ami a "kocsivissza" (Carriage Return) kódja.

   Például a következő, assembly nyelvű utasítással építhető fel egy paraméter mező:

```
DB 17,"\C:DISKCOMP A: B:",13
```

   Ez azt eredményezi, hogy a felhasználói parancsprocesszor betöltése után az végrehajtja a DISKCOMP programot.

3. Az EXEC függvényhívással (hexadecimális 4B) az új parancsprocesszor kerül végrehajtásra. Ez a parancsprocesszor-váltás tükröződik a programkörnyezetben is, amelynek címe PSP + 26, ez a cím egyben a COMSPEC mező kezdetére mutat. (A programkörnyezet, a PSP - program szegmens prefix - és a COMSPEC fogalmát a 15. fejezetben, a program szegmens leírásánál ismertejtük.)

4. A másodlagos parancsprocesszor "örökli" az első PROMPT parancsban megadott készenléti jelet. Visszatéréskor ez nem történik meg.

6.2 EXIT - visszatérés az elsődleges parancsfeldolgozóhoz

A COMMAND paranccsal meghívott másodlagos parancsfeldolgozóból való visszatérésre biztosít lehetőséget az EXIT parancs.

Verzió: 2.11 - 3.30
A parancs típusa: Belső
A parancs formája:

EXIT

Megjegyzés:

1. A parancsnak csak akkor van hatása, ha a COMMAND parancs kiadásakor sem a /S, sem a /P paramétert nem adtuk meg.

6.3 CLS - képernyő törlése

Törli a képernyő tartalmát. A kurzor a bal-felső pozícióba (home) kerül. A háttér és a keret színe nem változik meg.

Verzió: 2.11 - 3.30
A parancs típusa: Belső
A parancs formája:

CLS

Megjegyzés:

Ha a képernyő grafikus módra van kapcsolva, csíkozott lesz. Ilyenkor a képernyőt normál módba a MODE paranccsal lehet visszakapcsolni.

6.4 DATE - dátum megadása

A parancs lehetővé teszi a napi dátum megadását és változtatását. A pontos dátum megadása azért lényeges, mert a file-ok létrehozásakor, módosításakor az is bekerül a tartalomjegyzékbe a file azonosítójával együtt.

Verzió: 2.11 - 3.30
A parancs típusa: Belső
A parancs formája:

DATE [*hh-nn-éé*] | [*nn-hh-éé*] | [*éé-hh-nn*]

A paraméterek jelentése:

hh a hónap sorszáma, értéke: 1-12.

nn a hónapon belüli nap sorszáma, értéke: 1-31.

éé az év azonosítója, amely két formában adható meg:

- ha két számjegyet adunk meg, akkor 19-t feltételez, értéke ekkor: 80-99,
- négy számjegyet is megadhatunk, ekkor az értéke: 1980-2099.

Megjegyzés:

1. A paramétereket elválasztó jel / (slash) is lehet.
2. Ha érvénytelen paramétert adunk meg, akkor a rendszer ismételten kéri az új karaktersorozatot.

3. Ha csak a DATE kulcsszót írjuk be, akkor a DOS kiírja a rendszer-dátumot, majd kéri az aktuális dátumot. Ha nem akarjuk változtatni, akkor csak az üzenet végét jelző billentyűt (< ENTER >) kell lenyomni.

4. A dátum megadási módja a COUNTRY konfiguráció, illetve a SELECT DOS parancs használatától függ.

Példa:

Adjuk meg az új dátumot, ha az 86.03.07.

```
C > DATE

Date is: Thu 6-3-1986
Enter new date: 7/3/86
```

6.5 TIME - idő megadása

A parancs lehetővé teszi a pontos idő megadását, változtatását. A rendszer-idő a rendszer-dátumhoz hasonlóan bekerül a file létrehozásakor, módosításakor a tartalomjegyzékbe.

Verzió: 2.11 - 3.30
A parancs típusa: Külső
A parancs formája:

TIME [*óó*[:*pp*[:*mm*[.*xx*]]]]

A paraméterek jelentése:

óó az óra megadása, értéke: 0-23.
pp a perc megadása, értéke: 0-59.
mm a másodperc megadása, értéke: 0-59.
xx a századmásodperc megadása, értéke: 0-99.

Megjegyzés:

1. Ha érvénytelen paramétereket adunk be, akkor a rendszer ismételten kéri az új időt.

2. Ha a századmásodperc részt nem adjuk meg, akkor 00-t feltételez.

3. Ha csak a TIME kulcsszót adjuk meg, akkor a DOS kiírja a rendszeridőt, majd kéri a pontos időt. Ha nem akarjuk változtatni, akkor csak az üzenet végét jelző billentyűt (< ENTER >) kell lenyomni.

Példa:

Adjuk meg a pontos időt, ha az: 8:10:37.00

```
C>TIME

Time is: 00:03:54.65
Enter new time: 8:10:37
```

6.6 VER - DOS-verzió lekérdezése

A parancs kiírja a DOS verziószámát.

Verzió: 2.11 - 3.30
A parancs típusa: Belső
A parancs formája:

VER

Megjegyzés:

A DOS-verzió a következő információkat tartalmazza:

- a forgalmazó megnevezése,
- a főverziószám,
- a főverzión belüli változat.

Ha a parancsot az IBM DOS rendszere alatt adjuk ki, akkor IBM Personal Computer DOS Version 3.30 jelenik meg a képernyőn, az SZKI PROPOS-16 rendszerében pedig SZKI PROPOS-16 Ver. 3.xx

6.7 ASSIGN - periféria-hozzárendelések megváltoztatása

A parancs lehetővé teszi, hogy egy másik meghajtót jelöljünk ki a lemezes I/O műveletekhez.

Verzió: 2.11 - 3.30
A parancs típusa: Külső
A parancs formája:

ASSIGN [*x*[=]*y* [...]]

A paraméterek jelentése:

Az *y* meghajtót *x*-szé nevezhetjük át. A meghajtó megadása után nem kell kettőspontot (:) tenni.

Megjegyzés:

1. Ha a parancsot paraméterek nélkül adjuk meg, akkor az eredeti meghajtó hozzárendelések visszaállnak.

2. A parancs alkalmazása csak olyan felhasználói programok futtatása előtt ajánlott, amelyekbe "be van építve", hogy egy megadott meghajtót használjanak, de az most nem áll rendelkezésünkre.

3. Ez a parancs nem az input vagy output átirányítását jelenti, hanem a meghajtó ideiglenes átnevezését.

4. A FORMAT, DISKCOPY és DISKCOMP parancsok figyelmen kívül hagyják a parancs hatását.

5. A parancsot nem lehet együtt használni a BACKUP, RESTORE, LABEL, JOIN, SUBST és PRINT parancsokkal.

Példa:

1. A következő példában az A: és a B: meghajtóról valamennyi I/O műveletet a fix lemezre irányítunk.

ASSIGN A=C B=C

2. A következő parancs, az

ASSIGN

az eredeti hozzárendeléseket visszaállítja.

6.8 BREAK - < Break > billentyű figyelése

Ez a parancs lehetővé teszi, hogy a rendszer a DOS-funkciók elvégzése alatt figyelje, hogy a < Ctrl-Break > billentyűt lenyomtuk-e.

Verzió: 2.11 - 3.30
A parancs típusa: Belső
A parancs formája:

BREAK [ON | OFF]

A paraméterek jelentése:

Ha az **ON** paramétert adjuk meg, akkor a < Ctrl-Break > bármilyen DOS funkció hívásakor érvényre jut.

Ha az **OFF** paramétert adjuk meg, akkor a < Ctrl-Break > csak akkor kerül érvényre, amikor a végrehajtás alatt álló program a billentyűzetre, a képernyőre, a nyomtatóra vagy valamelyik vonali adapterre kér I/O műveletet. Ez a BREAK kapcsoló alapértelmezése.

Ha nem adunk meg paramétert, akkor aktuális értéke kerül kijelzésre.

Megjegyzés:

1. A BREAK értéke mindaddig érvényben marad, amíg egy másik BREAK parancscsal azt át nem állítjuk.

2. A BREAK ON megadása csak akkor indokolt, ha egy olyan programot futtatunk, amely főleg file I/O műveleteket végez, de a felhasználóval alig kommunikál (nincs/kevés a billentyűzet input, ill. képernyőre írás).

6.9 MODE - perifériák működésmódjának beállítása

A parancs lehetőséget ad a képernyő, a nyomtató és az aszinkron kommunikációs adapterek működésmódjának a beállítására. Alkalmazásának területei:

- a képernyő üzemmódjának beállítása,
- a nyomtatási formátum meghatározása,
- az aszinkron kommunikációs adapter üzemmódjának a beállítása,
- a párhuzamos illesztésű nyomtató outputjának átirányítása az aszinkron kommunikációs adapterre.
- a kódlapozás paramétereinek beállítása,

Ha a MODE parancsot nem a képernyő üzemmódjának beállítására használjuk, akkor a DOS mérete megnő.

Verzió: 2.11 - 3.30
A parancs típusa: Külső
A parancs formája:

1. **MODE LPT#**[:][*n*][,[*m*][,**P**]]
2. **MODE** *n*
3. **MODE** [*n*],*m*[,**T**]
4. **MODE COM***n*[:]*baud*[,[*paritás*][,[*adatbitek*][,[*stopbitek*][,**P**]]]]
5. **MODE LPT#**[:]=**COM***n*
6. **MODE** *egység* **CODEPAGE PREPARE**=((*kp*)[*d:*][*út*]*file-név*[*.kit*])
7. **MODE** *egység* **CODEPAGE PREPARE**=((*kp-lista*)[*d:*][*út*]*file-név*[*.kit*])
8. **MODE** *egység* **CODEPAGE SELECT**=*kp*
9. **MODE** *egység* **CODEPAGE** [**/STATUS**]
10. **MODE** *egység* **CODEPAGE REFRESH**

Az egyes kulcsszavakat a következőképpen rövidíthetjük:

- CP CODEPAGE helyett
- PREP PREPARE helyett
- SEL SELECT helyett
- REF REFRESH helyett
- STA STATUS helyett

1. Alkalmazás - A nyomtató üzemmódjának beállítása.

A parancs formája:

MODE LPT#[:][*n*][,[*m*][,**P**]]

A paraméterek jelentése:

Értéke 1, 2, illetve 3 lehet.
n Az értéke 80, vagy 132 lehet,az egy sorba nyomtatható karakterek maximális száma.
m Az értéke 6 vagy 8 lehet, a sorok száma 1 coll (= 2.54 cm) távolságon.
P A folyamatos ismétlést állítja be idő-túllépés (time-out) hiba esetén. Ha P-t adtunk meg, és már nem akarjuk, hogy a rendszer a time-out hibánál újra próbálkozzon, a P hatását úgy lehet megszüntetni, hogy a nyomtatót újra inicializáljuk a P megadása nélkül.

Megjegyzés:

A nyomtatóra bekapcsoláskor a következő alapértelmezések élnek: l = 80; m = 6; tehát 80 karakter soronként és 6 sor/coll.

Példa:

A következő paranccsal a nyomtató üzemmódjának valamennyi paraméterét állítjuk:

MODE LPT1:132,8,P

Tehát 132 karakter/sor; 8 sor/coll; idő-túllépés (time-out) esetén pedig folyamatos ismétlés.

II. Alkalmazás - A színes/grafikus képernyő adapter üzemmódjának a beállítása, átváltása.

A parancs formája:

1. **MODE** *n*
2. **MODE** [*n*],*m*[,T]

A paraméterek jelentése:

n Értéke: 40; 80; BW40; BW80; CO40; CO80; vagy MONO lehet.
40 A színes/grafikus adapter esetén az egy sorban kijelezhető maximális karakterek száma: 40.
80 A színes/grafikus adapter esetén az egy sorban kijelezhető maximális karakterek száma: 80.
BW40 Az aktív képernyő adaptert fekete-fehér módra kapcsolja át, a kijelzés soronként 40 karakter lesz.
BW80 Ez ugyanaz, mint a BW40, csak soronként 80 karakterrel.
CO40 Az aktív képernyő adaptert színes módra kapcsolja át.
CO80 Ez ugyanaz, mint CO40, csak soronként 80 karakterrel.
MONO Az éppen aktív képernyő adaptert monokróm adapterre kapcsolja át. A képernyőre mindig 80 karakter írható ki soronként.
m Az értéke **R** vagy **L** lehet. A képernyő kijelzésének jobbra vagy balra történő mozgatását jelenti. (R - jobbra, L - balra.)
T A rendszer a képernyő beállításához teszt-képernyőformákat küld, amellyel a kijelzés ellenőrizhető. (Ez egyjegyű számokból álló sor.)

Megjegyzés:

1. A 40; 80; CO40; CO80 és a MONO értékek csak színes/grafikus adapter esetében használhatóak.

2. A képernyő kijelzése mindkét irányba eltolható:

   - 40 karakter/sor esetén 1 karakterrel,
   - 80 karakter/sor esetén 2 karakterrel.

   Ha a T paramétert megadjuk, akkor üzenetet kapunk, hogy megfelelő-e a képernyő. Ha az Y-t, adjuk meg, akkor a parancs befejeződik. Ha pedig az N-t, akkor a képernyő eltolását megismétli, és ismét üzenetet küld a rendszer.

Példa:

A következő parancs

MODE 80,R,T

esetén az üzemmód a következő:

- 80 karakter/sor,
- a képernyő kijelzését 2 karakterrel jobbra tolja,
- teszt-képernyőformát küld.

III. Alkalmazás - Az aszinkron kommunikációs adapter üzemmódjának beállítása.

A parancs formája:

MODE COM*n*[:]*baud*[,[*paritás*][,[*adatbitek*][,[*stopbitek*][,P]]]]

A paraméterek jelentése:

n	Értéke 1 vagy 2 (3.30 esetén még 3 vagy 4 is), az aszinkron kommunikációs adapter azonosítója.
baud	110; 150; 300; 600; 1200; 2400; 4800; 9600 (illetve 3.30 esetén 19200) átviteli sebesség. Az azonosítóhoz elég az első két karaktert megadni.
paritás	N - nincs paritás, O - páratlan, E - páros (alapértelmezés).
adatbitek	7 (alapértelmezés) vagy 8.
stopbitek	1 vagy 2. Ha az átviteli sebesség értéke 110 baud, akkor az alapértelmezés: 2, egyébként 1.

A P paramétert csak akkor kell megadni, ha az aszinkron adaptert egy soros illesztésű nyomtató esetében használjuk. Ha a P paramétert megadtuk, akkor az idő-túllépés (time-out) hiba esetére a folyamatos ismétlést állítottuk be. Ezt a <Ctrl>-<Break> billentyű lenyomásával szüntethetjük meg. A hibaüzeneteket úgy szüntethetjük meg, hogy a MODE parancsot kiadjuk a P paraméter nélkül.

Megjegyzés:

Az aszinkron adatátviteli-protokoll paramétereinek a megadása az adapter inicializálását is jelenti. Az átviteli sebesség értékét mindig meg kell adni.

Példa:

Ha a következő parancsot adjuk meg:

MODE COM1:12,N,8,1,P

akkor a következő értékekkel inicializáljuk az első aszinkron átviteli adaptert:

- átviteli sebesség: 1200 Baud,
- paritást nem ad,
- az adatbitek száma: 8,
- egy stopbit,
- idő-túllépés esetén ismétlés.

Az alapértelmezések felhasználásával ugyanez a parancs a következő formában is kiadható:

MODE COM1: 12,N,8,,P

IV. Alkalmazás - A párhuzamos illesztésű nyomtató átirányítása az aszinkron kommunikációs adapterre.

A parancs formája:

MODE LPT#[:] = COM*n*

A paraméterek jelentése:

1, 2 vagy 3, a nyomtató azonosítója,

n 1 vagy 2 (3.30 esetén még 3 vagy 4 is), az aszinkron adapter azonosítója.

Megjegyzés:

1. A párhuzamos illesztésű nyomtató outputjának átirányítása előtt az aszinkron kommunikációs adaptert inicializálni kell (III. Alkalmazás). Az inicializáló parancsnak a P paramétert is tartalmaznia kell.
2. Ha a MODE LPT#[:][n][,[m],P] parancsot kiadjuk (az n-dik nyomtató üzemmódjának beállítása), akkor az érvényteleníti az ugyanarra a nyomtatóra kiadott LPT# = COMn parancsot.

V. Alkalmazás - kódlapok előkészítése

A parancs formája:

MODE *egység* **CODEPAGE PREPARE =** ((*kp*)[*d:*][*út*]*file-név*[*.kit*])
MODE *egység* **CODEPAGE PREPARE =** ((*kp-lista*)[*d:*][*út*]*file-név*[*.kit*])

A paraméterek jelentése:

egység a következőlogikai egységek valamelyike: CON, PRN, LPT1, LPT2 vagy LPT3.
kp valamelyik kódlap sorszáma
kp-lista kódlapok sorszámainak listája. Az egyes értékeket vesszővel kell elválasztani és az egész listát kerek zárójelek közé kell tenni. (A kódlap csak a 437, 850, 860, 863 vagy 865 értékek valamelyike lehet.)

[*d:*][*út*]*file-név*[*.kit*] a kódlapokat tartalmazó file neve. A rendszerhez az alábbi CPI kiterjesztésű file-okat kapjuk:

- 4201.CPI - IBM Proprinter
- 5202.CPI - IBM Quietwriter III. Printer
- EGA.CPI - EGA kompatibilis eszköz
- LCD.CPI - IBM LCD

Példák:

MODE LPT1:CODEPAGE PREP = ((850,865) 4201,CPI)
MODE LPT1:CP PREP = ((437)D:\4201.CPI)
MODECON:CP PREP = ((850,,437) EGA.CPI)

A harmadik példában a 850-es és 437-es kódlap közé egy üres lap került. Ennek az a hatása, hogy a CON: egységre előzőleg definiált második kódlap változatlan marad.

VI. Alkalmazás - Kódlap kiválasztása

Az V. alkalmazásban csak a memóriába töltjük azokat a kódlapokat, amiket használni akarunk. Ezek közül a használni kívántakat az aalábbi paranccsaal lehet kiválasztani:

MODE *egység* **CODEPAGE SELECT =** *kp*

A paraméterek jelentése:

egység A következő logikai egységek valamelyike: CON, PRN, LPT1,LPT2 vagy LPT3.
kp a használni kívánt kódlap sorszáma. A lehetséges értékek: 437, 850, 860, 863 vagy 865. Csak olyan értéket választhatunk, amelyet egy megelőző MODE PREP parancs listájában felsoroltunk.

Példák:

MODE CON: CP SEL = 850
MODE LPT1: CP SEL = 865

VII. alkalmazás - Kódlap sorszámának lekérdezése

MODE *egység* **CODEPAGE [/STATUS]**

A parancs az egységhez rendelt kódlapok számát és az éppen aktív kódlap számát a képernyőre írja. 'egység' a (CON,PRN,LPT1, LPT2 és LPT3 logikai egységek bármelyike lehet.

VIII. alkalmazás - kódlap újratöltése

MODE *egység* **CODEPAGE REFRESH**

A parancs újra olvassa az utolsó MODE CP PREP parancsban megadott file-t. Erre időnként szükség lehet, ha pl. valamiért kikapcsoltuk a nyomtatót. Az 'egység' paraméter a CON, PRN, LPT1, LPT2 es LPT3 logikai egység bármelyike lehet.

Példa:

```
MODE LPT1: CP PREP = ((860) D:\CPDIR5202.CPI)
...
MODE LPT1: CP REF
```

6.10 KEYB - nemzeti billentyűzet kiválasztása

A parancs lehetővé teszi, hogy a DOS BIOS billentyűzet-kezelő programját felülbíráljuk, és a tasztatúrának megfelelő programot töltsünk be.

Verzió: 2.11 - 3.30
A parancs típusa: Külső
A parancs formája (3.20-as vagy korábbi verziónál):

KEYB*xx*

A parancsnak nincsenek paraméterei, de a kulcsszóban az *xx* helyére a következő karakterek kerülhetnek:

UK - angol,
FR - francia,
SP - spanyol,
GR - német,
IT - olasz.

A parancs formája (3.30 verziónál):

KEYB [*xx*[,[*yyy*],[[*d:*][*út*]*file-név*[*kit*]]]]

A paraméterek jelentése:

xx a használható billentyűzet kódok egyike.

yyy	a használandó kódlap. Ha nem adjuk meg, akkor a program az adott nemzeti billentyűzet kódhoz tartozó kódlappal dolgozik.
[d:][út]file-név[.kit]	a billentyűzet leíró file-t azonosítja. Ha nem adjuk meg, akkor a KEYB parancs az aktuális egység főtartalomjegyzékében levő KEYBOARD.SYS nevű file-t használja.

Megjegyzés:

1. A nemzeti billentyűzet kiválasztására szolgáló program 3.30-as verziója nem kompatibilis az előző verziókkal.
2. A DOS indítása után az aktuális klaviatúra az amerikai (feltéve hogy pl. az AUTOEXEC.BAT file-ban nem állítjuk át). Ha több KEYBxx parancsot adunk ki, akkor mindig újabb program töltődik a memóriába, de az előző sem törlődik.
3. A KEYBxx programból bármikor visszatérhetünk az amerikai billentyűzetre a <Ctrl-Alt-F1> lenyomásával.
4. Mindig az utoljára betöltött programot és az amerikai billentyűzet-kiosztást lehet csak elérni. A rendszer újraindításakor a betöltött tasztatúra-kezelő programok törlődnek.
5. Magyarországon több tasztatúra-kezelő program is megvalósítja a magyar ékezetes (szabvány írógép) billentyűkezelést. Közülük néhányat KEYBHU-nak neveznek.

6.11 GRAFTABL - grafikai jelek betöltése

A parancs segítségével a színes/grafikus adapter működéséhez egy karaktertáblázatot tölt be a memóriába. A táblázat a memóriában a PC DOS rendszer alatt 1392 byte-ot foglal el.

Verzió: 2.11 - 3.30
A parancs típusa: Külső
A parancs formája:

GRAFTABL [437|860|863|865|/STATUS]

A paraméterek jelentése:

437	- USA (alapértelmezés)
860	- Portugál
863	- Kanadai (francia)
865	- Norvég vagy dán
/STATUS	hatására kiírja a kiválasztott kódlap számát.

Megjegyzés:

1. A parancs az 1F megszakítási vektort (interrupt vektor) változtatja meg, így a 128 és a 255 érték közötti ASCII-kódú jelek is kiírathatók a képernyőre, ha a színes/grafikus adaptert grafikus módba állítottuk be.

2. A parancs a kódtáblázat betöltése és a megszakítási vektor átállítása után a következő üzenetet adja:

   GRAPHIC CHARACTERS LOADED
   (A grafikus jelkészletet betöltötte.)

3. A parancs ismételt kiadása esetén a rendszer a következő üzenetet küldi:

   GRAPHIC CHARACTERS ALREADY LOADED
   (A grafikus jelkészletet már betöltötte.)

   Ez a parancs a Propos 3.0 és a Commodore PC-10, PC-20 operációs rendszerében nem áll rendelkezésre.

4. A DOS néhány verziójánál, ha a GRAFTABL parancsot a rendszer indítása után ismételten kiadjuk, nem kapunk üzenetet. A parancs ismételt kiadása után viszont a DOS mérete mindig mintegy 132 Kbyte-tal bővül (így a felhasználói szabad memória ennyivel csökken).

   A GRAFTABL a következő kódokkal tér vissza:

   0 - Nem volt előzően definiált kódlap, a megadott rezidensen a tárban maradt.
   1 - Volt már előzően definiált kódlap, ha az új attól eltérő, akkor az előző helyére kerül.
   2 - Nem volt előzőleg definiált kódlap és nem is keletkezett.
   3 - Nem megfelelő paraméter
   4 - Nem megfelelő DOS verzió.

6.12 GRAPHICS - grafikus nyomtatás lehetősége

Ez a parancs lehetővé teszi, hogy a képernyő grafikus tartalmát a nyomtatóra kiírassuk. A parancs kiadása után a DOS rezidens része a memóriában 688 byte-tal megnő. Ez a parancs csak akkor használható, ha a számítógép rendelkezik színes/grafikus adapterrel és monitorral.

Verzió: 2.11 - 3.30
A parancs típusa: Külső
A parancs formája:

GRAPHICS [*nyomtató típusa*][**/R**][**/B**][**/LCD**]

A paraméterek jelentése:

/R esetén ami a képernyőn fehér, az nem kerül nyomtatásra, ami fekete, az igen. Ha a paramétert nem adjuk meg, akkor a fehér fekete, a fekete pedig fehér lesz.

/B megadásának csak színes nyomtató esetén van értelme, s hatására a háttér színét is nyomtatja a rendszer. Egyébként a háttér színe fehér lesz nyomtatáskor.

/LCD megadása után pontosan úgy nyomtat, ahogy az az IBM PC Convertible Liquid Crystal Display típusú képernyőn megjelenik.

A 'nyomtató típusa' lehetséges értékei a következőek:

COLOR1 - IBM színes nyomtató fekete szalaggal;
COLOR4 - IBM színes nyomtató RGB szalaggal;
COLOR8 - IMB színes nyomtató CMY szalaggal;
COMPACT - IBM Compact Printer;
GRAPHICS - IBM Personal Graphics Printer
THERMAL - IBM PC Convertible Printer és IBM Proprinter

Ha nem adjuk meg, az GRAPHICS-nek számít. Ezzel a fajtával kompatibilisek az EPSON FX sorozatú nyomtatók.

Megjegyzés:

A képernyő kinyomtatását a <Shift-PrtSc> billentyűk lenyomásával indíthatjuk.

Ha a képernyő grafikus módban van, a <Shift-PrtSc> minden egyes lenyomására a következő történik:

- 320 x 200-as színes/grafikus módban a képernyő tartalmát négy szürke árnyalattal a nyomtatóra írja,
- 640 x 200-as színes/grafikus módban a képernyő tartalmát 90 fokkal elforgatva (az óramutató járásával ellentétes irányban) a nyomtatóra írja.

A képernyő teljes kinyomtatása legfeljebb 3 percig tart (természetesen függ a nyomtató típusától is).

A képernyő tartalmának kinyomtatása programból a következő módon kezdeményezhető:

```
PUSH BP
INT 5
POP BP
```

7. Parancs file-ok

A DOS lehetővé teszi, hogy parancsból paraméterezhető "eljárásokat" írjunk, ezeket file-okba tegyük, és innen hajtassuk végre. E file-okat itt **parancs-file**-oknak fogjuk nevezni (az angol irodalomban: **batch-file**). **Minden parancs-file nevének kiterjesztése .BAT.** A parancs-file-okban levő "eljárások" "utasításai" DOS parancsok lehetnek, ezek végrehajtását az e fejezetben sorra kerülő úgynevezett DOS alparancsokkal vezérelhetjük. A parancs file-ok végrehajtása a file-név és az aktuális paraméterek megadásával kezdeményezhető.

A <Ctrl-Break> billentyű lenyomásával a végrehajtás alatt álló DOS parancs feldolgozása befejeződik. A rendszer üzenetet küld, hogy a parancs-file további feldolgozását befejezze-e.

Ha Y-t adunk meg, akkor a parancs file további parancsait átlépi, figyelmen kívül hagyja, és a DOS prompt jelenik meg. Ha N-t adunk meg, akkor áttér a parancs-file következő parancsainak feldolgozására. Ha bármilyen egyéb karakter (sorozat) a válasz, akkor a DOS megismétli az üzenetet.

Megjegyzés:

1. A parancs-file-ok tetszőleges szerkesztő programmal (pl. EDLIN) vagy a COPY CON paranccsal hozhatók létre.
2. A parancs-file-t tartalmazó lemezt a végrehajtás során nem lehet kivenni. Ha ez megtörténik, a DOS üzenetet küld, melyben kéri a felhasználót, hogy a lemezt helyezze vissza a meghajtóba.
3. A parancs-file utolsó sora (rekordja) egy másik parancs-file azonosítója is lehet, így közvetlenül hívható a következő parancs-file.
4. A parancs-file végrehajtása során az aktuális tartalomjegyzék tetszés szerint változtatható, mivel a rendszer feljegyzi, hogy a file melyik tartalomjegyzékben található.
5. A DOS 3.30 verziója alatt már lehetőség van egyik parancsfile-ból másikat úgy aktivizálni, hogy annak lefutása után a parancsok feldolgozása a hívó parancs-file-ban (az aktivizáló parancs után) folytatódik.

7.1 A parancs-file-ok meghívása és paraméterezése

A parancs-file meghívásának, azaz a benne foglalt parancsok és alparancsok végrehajtatásának általános alakja:

[*d:*][*út*]*file-név*[**.BAT**] [*paraméterek*]

Látható, hogy a kiterjesztés kötelezően .BAT.

7.1.1 Formális paraméterek

A file-ban a %0-%9 azonosítóval jelzett formális paraméterek adhatók meg. A %0 fenntartott paraméter, mivel ennek az értéke az eljárás végrehajtása alatt a parancs-file meghajtójának azonosítója (ha ez meg volt adva), és a parancs-file neve. A formális paraméterek az eljárásban a meghajtók és file-ok azonosítói helyén fordulhatnak elő.

Példa:

Legyen az ASMCOPY.BAT parancs-file-ban leírt eljárás:

```
COPY %1.MAC %2.MAC
COMP %1.MAC %2.MAC
TYPE %1.MAC
```

Ha az eljárást a következő módon indítjuk:

```
ASMCOPY A:FIRST1 B:FIRST2
```

akkor a következő parancsok hajtódnak végre:

```
COPY A:FIRST1.MAC B:FIRST2.MAC
COMP A:FIRST1.MAC B:FIRST2.MAC
TYPE A:FIRST1.MAC
```

A paraméterek értéke a hívás után a következő volt:

%0 - mindig a meghajtó- és a parancs-file azonosítóját tartalmazza. (Nem kell megadni)
%1 - a FIRST1 karaktereket tartalmazza
%2 - a FIRST2 karaktereket tartalmazza

Megjegyzés:

Abban az esetben, ha a parancs-file-ban egy olyan file-azonosító szerepel, amelynek a százalék (%) karakter része, akkor a százalékot kettőzve kell megadni.

Például:

Az ASM%MAC.MAC egy megengedett azonosító. Ha ezt egy parancsfile-ban akarjuk megadni, akkor ehelyett az ASM%%MAC.MAC azonosítót kell beírni.

7.1.2 Környezeti változók használata

A parancs-file-ban a SET segítségével beállítható/lekérdezhető környezeti változók is használhatók. Ezek értékére nevüknek % jelek közé zárásával hivatkozhatunk:

%prompt%

prompt tetszőleges, a SET parancsban előfordulható környezeti változó neve. Például a PATH környezeti változóra az alábbi formában hivatkozhatunk: **%PATH%**

Példa:

```
IF %cel% = = c: GOTO :fix
ECHO Tegyen be egy lemezt!
PAUSE
:fix
program > %cel%mentes.txt
```

7.2 Az AUTOEXEC.BAT file

A DOS indításakor azon a lemezen, ahonnan indítottuk, a parancsfeldolgozó program megkeresi a főtartalomjegyzékben az AUTOEXEC.BAT file-t, és automatikusan végre is hajtja. Ide betehetjük azokat a parancsokat, amelyeket a DOS minden indításakor és újraindításakor végre akarunk hajtatni.

Például:

A > COPY CON AUTOEXEC.BAT < ENTER >	a file létrehozása
PROMPT pg < ENTER >	a DOS prompt ennek hatására az aktuális meghajtó betűjele lesz, a > karakterrel követve (ld. PROMPT parancs)
PATH \ < ENTER >	beállítja a főtartalomjegyzéket (ld. PATH parancs)
DATE < ENTER >	bekérjük a dátumot
TIME < ENTER >	és időt
< F6 >	Ctrl + Z: file-vége jel
< ENTER >	

Ha a parancs-file írásakor hibázunk, adjunk < Ctrl-Break > -et. Ez a COPY-t anélkül fejezi be, hogy a file létrejönne. Ekkor újra kezdhetjük a file létrehozását.

Az egyes parancsokat < ENTER > -rel, a file írását az < F6 > majd < ENTER > gombokkal fejezzük be. Ennek hatására a COPY kiüzeni:

```
1 File(s) copied
```

Amikor gépünkön dolgozni kezdünk, nézzük meg, van-e már AUTOEXEC.BAT file (pl. a DIR paranccsal), ha nincs, a fenti módon hozzuk létre. Ha van, listázzuk ki (TYPE parancs), és ha a fenti négy parancsot nem tartalmazná, érdemes ezeket is megadni.

7.3 DOS alparancsok

Az ebben a szakaszban ismertetésre kerülő alparancsok csak parancs-file-okban szerepelhetnek. Alkalmazásukkal a parancs-file-ok feldolgozását lehet vezérelni.

Például:

- A parancs-file-ban parancsok átlépése a végrehajtás során;

- Egy parancs többszöri végrehajtása;
- A DOS parancsok kijelzése a képernyőre a végrehajtás során;
- Üzenetek küldése az operátornak.

Valamennyi alparancs típusa belső.

7.3.1 REM - megjegyzés

Ennek az alparancsnak a segítségével a parancs-file-ban megjegyzést lehet elhelyezni. A megjegyzés tetszőleges karaktersorozat lehet, amely emlékeztető, figyelmeztető stb. utalás lehet egy későbbi felhasználás során. Az alparancsot megjegyzés nélkül is lehet használni. A REM alparancs a parancs-file tagolását segíti.

A parancs formája:

REM [*megjegyzés*]

7.3.2 ECHO - parancsok megjelenítése

Ez az alparancs lehetővé teszi, hogy a parancs-file parancsainak a képernyőn való megjelenítése felől döntsünk. Az ECHO a rendszerüzenetek kiírását nem befolyásolja.

A parancs formája:

ECHO [**ON** | **OFF** | *üzenet*]

A paraméter jelentése:

A rendszer indítása után a parancs-file-ok végrehajtásra kerülő parancsai a képernyőn megjelennek, azaz az **ECHO ON** (bekapcsolt) állapotban van.

Az **ECHO OFF** segítségével ez az üzemmód megváltoztatható, ekkor egyik parancs sem jelenik meg a képernyőn.

Ha az ECHO "üzenet" paraméterrel szerepel a parancs-file-ban, akkor az "üzenet" mindig megjelenik a képernyőn, függetlenül attól, hogy az ECHO ON vagy OFF állapotban van-e.

Ha az ECHO parancsait paraméterek nélkül adjuk meg, akkor pillanatnyi értéke jelenik meg: ON vagy OFF.

Példa:

Egy parancs-file részletét mutatjuk be az ECHO alparancs alkalmazására:

Legyen a PARANCSF.BAT azonosítójú parancs-file tartalma:

```
ECHO BEGIN PF
ECHO OFF
DIR C:AUTO*
```

```
ECHO ON
DIR C:AUTO*
ECHO ENDECHO
```

Ekkor a PARANCSF utasítás hatására a képernyőn a következő jelenik meg:

```
C:\ > echo begin pf
begin pf
C:\ > echo off
```

A DIR parancs kiírását az ECHO OFF állapot sem tiltja le, de maga a parancs nem látható. A file-ok kiírása után még a következő üzenetet kapjuk:

```
C:\ > echo endecho
endecho
```

Megjegyzés:

Ha a parancs-file kiírásait úgy akarjuk szeparálni, hogy üres sorok is legyenek, akkor nem használhatjuk a paraméter nélküli ECHO parancsot (az ui. az aktuális értéket adja vissza), hanem a sorba valamilyen nem látható (pl. 255 ASCII kód) karaktert kell helyezni. Ha a parancs file-ból a felhasználót hangjelzéssel is figyelmeztetni akarjuk, az ECHO üzenet sorába a 7 ASCII kódot kell beírni.

(Minden ASCII kód beadható a numerikus billentyűzetről az < Alt > billentyű lenyomása közben decimálisan beírva az értéket.)

7.3.3 PAUSE - feldolgozás megszakítása

Ez az alparancs lehetővé teszi, hogy felfüggesszük a rendszer működését. Erre akkor van szükség, ha valamilyen operátori tevékenységet, pl. lemezcserét meg kell a rendszernek várnia. Ha a tevékenységet elvégeztük, akkor egy tetszőleges billentyű lenyomásával a rendszer továbbindítható.

A PAUSE segítségével egy parancs-file PAUSE utáni részének további feldolgozása felől is dönthetünk. Mivel hatására a DOS megáll, meggondolhatjuk, szükség van-e a parancs további részének végrehajtására.

A parancs formája:

PAUSE [*üzenet*]

A paraméter jelentése:

Az *üzenet* legfeljebb 121 karakter. Ha a vezérlés a PAUSE-ra kerül akár parancs-file-ban van a PAUSE, akár közvetlen utasításként adtuk ki, kiírja először az "üzenet"-et, majd kiírja:

Strike any key, when ready...

Ez azt jelenti, hogy ha készen vagyunk (a gondolkodással vagy egyéb tevékenységgel),

nyomjunk le egy tetszőleges billentyűt.

Ha a PAUSE a parancs-file egy utasítása, és nem akarjuk a parancs-file további részét végrehajtani, adjunk <Ctrl-Break>-et. Ekkor megkérdezi:

Terminate batch job (Y/N)?

Azaz, befejezze-e a parancs-file végrehajtását. Ha Y-t adunk, befejezi, ha N-et, akkor nem. Egyéb karakter beírása esetén a kérdés megismétlődik.

Megjegyzés:

Természetesen az ECHO OFF utasítással megakadályozhatjuk az "üzenet" kiírását, de a "Strike any key..." és egyéb rendszer-üzenet továbbra is látható.

Példa:

A következő parancs-file részletben az operátort a nyomtató beállítására kérjük fel:

```
ECHO ON
PAUSE T. Operátor, ellenőrizd a nyomtatóban a papír beállítását. Ha kész, nyomj
meg egy tetszőleges billentyűt!
REM File-ok kiíratása a nyomtatón.
TYPE A:%2 >PRN
```

7.3.4 IF - feltételes végrehajtás

Ez az alparancs lehetővé teszi a parancs-file-beli DOS parancsok feltételes végrehajtását. A feltételek a következők lehetnek:

- az előző program befejezési kódjának értéke,
- két karaktersorozat összehasonlításának eredménye,
- megadott file létezése.

A parancs formája:

IF [**NOT**] *feltétel parancs*

A paraméterek jelentése:

A *feltétel* a következők egyike lehet:

- **ERRORLEVEL** *érték*
- *karaktersorzat1* = = *karaktersorozat2*
- **EXIST** *file-azonosító*

Ha a feltétel IGAZ, akkor a feltétel mögött álló DOS parancs végrehajtásra kerül (általában ez egy GOTO alparancs).

Az **ERRORLEVEL** *érték* feltétel akkor IGAZ, ha az előző program az "*érték*"-kel egyenlő, vagy annál nagyobb befejezési kódot (exit kód) állított be. Az érték decimális szám.

A *karaktersorozat1* == *karaktersorozat2* feltétel akkor IGAZ, ha a két sorozat minden pozícióban azonos jelet tartalmaz (egy nagybetűt nem tekint a rendszer a megfelelő kisbetűvel azonosnak).

Az **EXIST** *file-azonosító* feltétel akkor IGAZ, ha a rendszer a megadott file-t eléri. Megengedettek a globális név-karakterek (? és *).

A NOT a feltétel tagadását jelenti.

Példa:

1. A példában egy file létezését vizsgáljuk.

```
IF EXIST %1 GOTO G1
.
.
.
:G1
REM a file létezik, kiíratása következik
TYPE %1
```

2. A példában egy olyan program befejezését ellenőriztük, amelynél a sikeres befejezés esetén a befejezési kód = 0, sikertelen befejezés esetén pedig 1.

```
PROGRAM1
IF ERRORLEVEL 1 GOTO G1
PAUSE a program sikeresen befejeződött, cseréld le a lemezt az A: meghajtón.
GOTO G2
:G1
ECHO a PROGRAM1 hibás!
.
.
:G2 ...
```

3. A példában két karaktersorozat egyezését vizsgáljuk.

```
IF %1 = =A GOTO G2
.
.
:G2
REM az A: meghajtóra történik a mentés.
PAUSE a rendszer mentés lemezét tedd az A: meghajtóba. Ha kész, nyomj le egy
tetszőleges billentyűt.
BACKUP C: A:/M
REM mentésre kerültek a fix lemezről azok a file-ok, amelyek az utolsó mentés óta
változtak.
```

7.3.5 GOTO - ugrás a parancs-file-ban

A parancs-file-ban a parancsok végrehajtása a file-beli sorrendjükben történik. Ezt a sorrendet lehet a GOTO alparanccsal megváltoztatni.

A parancs formája:

GOTO *cimke*

A paraméter jelentése:

A *cimke* egy karakterekből álló jelsorozat. A cimkéket első nyolc karakterük különbözteti meg. A végrehajtás a parancs-file-ban a cimke utáni első DOS paranccsal folytatódik. Ha a végrehajtás sorrendjének megfelelően jut a vezérlés egy címkére, akkor a címke hatástalan.

Megjegyzés:

1. Ha a GOTO után írt címke nem található a file-ban, akkor üzenetet kapunk (label not found) és a végrehajtás befejeződik.
2. Az ugrások végrehajtásakor a címkék nem kerülnek kijelzésre. A címke után egy tetszőleges megjegyzés (lásd REM alparancs) írható. Ez segíti a parancs-file ellenőrzését, áttekintését.
3. A GOTO alparancs a parancs-file struktúráját rontja. Nem körültekintő alkalmazása esetén könnyen előáll a végtelen ciklus.
4. A *cimke*-t a file-ban bárhol elhelyezhetjük, a vezérlés az azt követő első parancsra adódik. A *cimke*-t tartalmazó sorban más nem lehet, s a *cimke* elé kettőspontot (:) kell rakni.

Példa:

Példánkban egy egyszerű végtelen ciklust mutatunk be.

```
:CIKLUS
ECHO ez egy végtelen ciklus!
GOTO CIKLUS
```

7.3.6 FOR - DOS parancsok ismételt végrehajtása

Ez az alparancs lehetővé teszi, hogy egy DOS parancsot többször végrehajtson a rendszer anélkül, hogy a parancsot ismételten megadnánk.

A parancs formája:

FOR *változó* **IN** (*mondat*) **DO** *parancs*

A paraméterek jelentése:

A "*változó*" egyenként felveszi a "*mondat*" egyes elemeinek értékét, és ezzel az értékkel a **DO** utáni "*parancs*" rendre végrehajtódik, amennyiben a parancs ezzel az értékkel értelmes. A "*változó*" formája:

%karakter

ahol *karakter* egy betű vagy egy számjegy, ha a FOR-t DOS utasításként, parancs-file-on kívül használjuk, és

%%karakter,

ahol *karakter* a fenti, ha a FOR parancs-file-beli utasítás.

A "*mondat*" legalább egy, (egymástól szóközökkel elválasztott) elemből áll. Az egyes elemekkel együtt a "*parancs*"-nak értelmesnek kell lennie.

A "*parancs*" az ismétlődő DOS parancs, paramétereivel együtt. Általában paramétereiben szerepel a "*változó*".

Megjegyzés:

1. A mondat elemeiben, ha azok file-nevek, a globális file-név jelek is használhatók.
2. A FOR alparancsban út nem adható meg.
3. A FOR alparancsok nem skatulyázhatók egymásba, azaz a DO-t nem követheti újabb FOR (ha megkíséreljük, kiírja: "FOR cannot be nested").

Példa:

1. A következő parancs szerepel a parancs-file-ban:

```
FOR %%IN (PROG1.ASM PROG2.ASM PROG3.ASM) DO
COPY A:%%IN B:
```

Ez a következő parancsokkal egyenértékű:

```
COPY A:PROG1.ASM B:
COPY A:PROG2.ASM B:
COPY A:PROG3.ASM B:
```

2. Próbáljuk ki a DOS prompt után a következő parancsot:

```
FOR %x IN (AUTOEXEC.BAT COMMAND.COM) DIR %x
```

Hatása a két file-ra külön-külön kiadott DIR paranccsal egyenértékű.

7.3.7 SHIFT - a paraméterek léptetése

Ez az alparancs lehetővé teszi, hogy a felhasználó tíznél több paramétert használjon (%0-%9 voltak a megengedett formális paraméterazonosítók).

A parancs formája:

SHIFT

A parancs működése

A formális paraméterek száma tíz. Ebből a %0 paraméter aktuális értéke mindig a meghajtó és a parancs-file-azonosító.

Ha a felhasználónak tíznél több paraétert kell használni, akkor SHIFT alparancsot alkalmazhat. Az alparancs hatására a parancssor minden paramétere eggyel balra tolódik, tehát a %0 új értéke a %1 régi értéke, %1 új értéke a %2 régi értéke stb.

Az alparancs akkor is használható, ha egy másik parancs-file-t hívunk meg a file utolsó utasításában.

Példa:

Készítsük el a következő PARANCSF.BAT parancsfile-t:

```
ECHO OFF
ECHO PARF KEZDET
TYPE %0.BAT
FOR %%X IN (%1 %2 %3 %4 %5 %6 %7 %8) DO TYPE %%X
PAUSE FOR UTAN a PAUSE utasításokat azért tesszük be, hogy egy képernyő teleírása után megálljon a rendszer
COPY %9 %8
TYPE %8
PAUSE SHIFT ELOTT
SHIFT
TYPE %9
ECHO PARF VEGE
```

Legyen a PR1, PR2, PR3, PR4, PR5, PR6, PR7, PR8, PR9 és PR10 file-unkban csak ez az egy sor:

```
EZ PRn
```

ahol n a megfelelő file nevében szereplő szám. Ezután hívjuk meg parancs-file-unkat a következő módon:

```
PARANCSF PR1 PR2 PR3 PR4 PR5 PR6 PR7 PR8 PR9 PR10
```

Ennek eredményeképpen megkapjuk először magának a parancs-file-nak, majd a többi file-unknak a listáját, összesen 11 db listát. Megjegyezzük, hogy a COPY végrehajtása után természetesen a PR8 tartalma:

```
EZ PR9
```

A többi file tartalma nem változik.

7.3.8 CALL - parancs-file hívása

A CALL alparancs segítségével a parancs-file-ban egy másik parancs-file végrehajtását rendelhetjük el, hasonlóan a szubrutin-híváshoz.

A parancs formája:

CALL [*d:*][*út*] *file-név*

8. A sorszerkesztő program (EDLIN)

A számítógép használata során gyakran lesz arra szükségünk, hogy ún. szöveges (text) file-okat állítsunk elő és tároljunk lemezen. Ilyenek például:

- magasszintű nyelvek (pl. PASCAL, FORTRAN) vagy az assembly nyelv forrásprogramjai,
- olvasásra, tájékoztatásra szánt szövegek (levelek, dokumentációk),
- számítógépes feldolgozásra kerülő file-ok (csak karakteres ábrázolású adatokkal),
- DOS parancs-file-ok.

Eddigi ismereteink szerint a COPY paranccsal tudunk egy szöveget a billentyűzetről a lemezre másolni, például így:

```
COPY CON MASOL.BAT < ENTER >
DISKCOPY A: B: < ENTER >
DISKCOMP A: B: < F6 > - < ENTER >
```

A fenti példában egy kétsoros parancs-file-t hoztunk létre, amelyet a MASOL paranccsal lehet behívni. Ez az eljárás egyszerűnek tűnik, ha a szöveg beírása közben hibát tartalmazó sort még nem zártuk le az < ENTER > -rel, mert visszaléphetünk a < Backspace > (visz-szatörlés) billentyűvel és javíthatunk. Ha azonban a hibás sort már elhagytuk, nincs lehetőségünk a korrigálásra, és újra be kell írnunk a teljes szöveget. A lemezen tárolt file-ok módosítására pedig a COPY utasítás végképp alkalmatlan.

Szerencsére azonban más módszer is kínálkozik a szöveges file-ok létrehozására és módosítására. Erre a célra speciális programok (editorok, szövegszerkesztők) állnak rendelkezésre, egyet közülük a DOS rendszerlemezen is megtalálunk, EDLIN néven. (A PROPOS-16 operációs rendszerben a hasonló szerkesztő program neve: EDIT.)

Az EDLIN (Line Editor, sorszerkesztő) program az alábbi feladatok végrehajtására képes:

- új szöveges file-ok létrehozása és tárolása lemezen.
- meglevő szöveges file-ok módosítása, az új és a régi változat megőrzésével,
- a szöveg egyes sorainak kiíratása, módosítása, törlése, másolása,áthelyezése, új sorok beszúrása,
- szövegrészek keresése, helyettesítése vagy törlése.

A felsorolásból látszik, hogy gyakorlatilag minden műveletet el lehet végezni a szövegeken az EDLIN segítségével. Más kérdés viszont, hogy mennyire kényelmes és hatékony ez a munka. Az EDLIN - a nevéből következően - csak sorszerkesztő, ezért az egyes sorok írása, módosítása közben nem léphetünk át másik sorba, csak ha előbb az adott sor szerkesztését befejeztük, < ENTER > -rel lezártuk. A művelet végrehajtásához külön parancsokat kell kiadni, és nem lehet a szövegben felfelé vagy lefelé lapozni (scroll-ozni). Aki a BASIC teljes képernyős editorához szokott, nehezen tudja elfogadni az EDLIN korlátait.

Ha hosszabb programokat, szövegeket kell írnunk, célszerű tehát az EDLIN helyett egy szövegszerkesztőt (például a WordStart) vagy egy teljes képernyős editort (például a Per-

sonal Editort) alkalmazni. Kisebb módosításokra vagy rövid szövegek írására viszont az EDLIN tökéletesen megfelel, kis mérete, rövid betöltési ideje, egyszerű és hatékony használata is előnyt jelent.

Az EDLIN által létrehozott szöveges file sorokra tagolódik, az egyes sorok váltakozó hosszúságúak lehetnek, de legfeljebb 254 karaktert tartalmazhatnak. A sorokat CR-LF (kocsivissza-soremelés) jelek zárják le. A sorok számát csak a file-t tartalmazó lemez kapacitása korlátozza; az EDLIN-nel az operatív memória méreténél hosszabb szöveget is fel lehet dolgozni.

Az ún. bináris file-ok (pl. az object- és a végrehajtható programokat tartalmazó file-ok) módosítására az EDLIN nem alkalmas. Erre a célra a DEBUG rendszerprogramot ajánljuk, melyet a 9.fejezetben ismertetünk.

8.1 Az EDLIN indítása

Az EDLIN program a DOS renszerlemezen található, EDLIN.COM néven, de szükség szerint átmásolhatjuk a szöveges file-t tartalmazó lemezre is. Külső program lévén fontos, hogy az EDLIN indításakor a programot tartalmazó lemez valamelyik meghajtóegységben legyen.

Az EDLIN indító parancsának általános formája:

EDLIN [*d:*][*út*]*file-név*[*.kit*][**/B**]

ahol [d:][út]file-név[.kit] a létrehozandó, illetve módosítandó file specifikációja.

/B-t akkor kell megadni, ha a file-on belül file-vége(<Ctrl-Z>) karakterek találhatók. (/B nélkül ugyanis az EDLIN az első file-vége jelnél abbahagyja a file beolvasását.)

Ha egy új szöveges file-t akarunk létrehozni, akkor az indító parancsban a lemezen nem szereplő file nevét kell megadnunk, például:

```
A>B:EDLIN SZOVEG.TXT
New file
*
```

A példában az EDLIN programot tartalmazó lemez a B egységben található, az új file-t (SZOVEG.TXT) pedig az aktuális meghajtón, az A egységen hozzuk létre. Az EDLIN a bejelentkezésekor kiírta: "New file" (új file), majd megjelent az EDLIN prompt-ja (felszólító jele), a csillag, amely azt jelenti hogy a program a parancsunkat várja. Az EDLIN a fenti indítás hatására megnyitja a SZOVEG.$$$ nevű ideiglenes file-t, amellyel a feldolgozás végén két dolog történhet. Ha tárolással fejezzük be a szerkesztést ("E" parancs), akkor a SZOVEG.$$$ megkapja a ténylegesen előírt kiterjesztést, és SZOVEG.TXT lesz belőle. Ha tárolás nélkül lépünk ki az EDLIN-ből ("Q" parancs), akkor pedig a SZOVEG.$$$ törlődik a lemezről.

Ha az EDLIN-nel egy már meglevő szöveges file-t akarunk módosítani, akkor az indító parancsba a megfelelő file nevét kell írnunk. Ebben az esetben szükséges lehet a /B opció használata is. Egy példa:

```
A>EDLIN B:MEGLEVO.DOC
End of input file
*
```

A programot az A egységről indítottuk, a MEGLEVO.DOC nevű file-t tartalmazó lemezt pedig a parancs kiadása előtt behelyeztük a B egységbe. Az "End of input file" (input file vége) üzenet azt jelzi, hogy az EDLIN a file teljes szövegét beolvasta a memóriába. Ha a szöveg nem fér el a rendelkezésre álló szabad memóriában, akkor az "End of input file" szöveg nem jelenik meg, csak az EDLIN prompt-ja, a csillag. Megjegyezzük, hogy az EDLIN csak egy 64kB-os memóriaterületet tud felhasználni, ezért a hosszabb szövegeket a nagy memóriakapacitású gépeken is csak részletenként lehet feldolgozni.

Meglévő file módosításakor az EDLIN az adatok biztonsága érdekében a következőképpen jár el: a file új, módosított változata számára először egy .$$$ kiterjedésű file-t nyit meg (példánkban a MEGLEVO.$$$-t), de az eredeti file-hoz nem nyúl. Ha a szerkesztést tárolással fejezzük be, akkor a régi eredeti file a .BAK kiterjesztést kapja (MEGLEVO.BAK), és a lemezen marad. A BAK rövidítés a backup-ra, biztonsági másolatra utal. Ha már volt a lemezen egy korábbi MEGLEVO.BAK file, akkor az az átnevezés előtt törlődik. Ezután az új, módosított file megkapja a végleges nevét (MEGLEVO.$$$ helyett a MEGLEVO.DOC nevet)

Ha a szerkesztésből tárolás nélkül lépünk ki (pl, ha hibásan hajtottuk végre a módosítást és újra akarjuk kezdeni az eredeti file-lal) akkor az EDLIN egyszerűen törli a .$$$ kiterjesztésű ideiglenes file-t és - mivel a régi file megmaradt- ezzel vissza is áll az eredeti állapot.

Az EDLIN ismertetett file-kezelési stratégiája következtében a módosított szövegek (például a fejlesztés alatt álló forrásprogramok) mindig két generációban állnak rendelkezésre: az egyik a legfrissebb, a másik az eggyel korábbi változat. E módszernek két további következménye: a .BAK kiterjesztésű file-okat nem lehet az EDLIN-nel feldolgozni (ebben az esetben megegyezne az eredeti és az archív file neve), ezért ilyenkor az illető file-t más névre kell átneveznünk. A másik következmény: a file-ok módosításakor az eredeti és az archív változat egyazon lemezen helyezkedik el, ezért együttes méretük nem haladhatja meg a lemezen rendelkezésre álló területet.

Egy példa a LEVEL.BAK nevű szöveg átszerkesztésére:

```
A>RENAME B:LEVEL.BAK LEVEL.TXT
A>EDLIN B:LEVEL.TXT
End of input file
*
```

A szerkesztés tárolással történt befejezése után a módosított szöveg a LEVEL.TXT-be, a régi szöveg pedig a LEVEL.BAK file-ba kerül.

8.2 Az EDLIN parancsai

Az EDLIN elindítása után egy csillag jelzi a képernyőn, hogy a program a parancsainkat várja. Ilyenkor EDLIN utasításokat kell beírnunk, amelyeknek semmi köze nincs a DOS

parancsaihoz. E speciális parancsokkal majd sorban megismerkedünk, de először nézzük meg a parancsok néhány általános elemét.

Az EDLIN sorszerkesztő, ezért a legtöbb parancsában utalnunk kell arra, hogy az adott műveletet mely soron vagy sorokon kell végrehajtania. Az EDLIN a sorokat 1-től kezdve, egyesével számozza. A sorszámok nem részei a soroknak, nem tárolódnak a lemezen, de a képernyőn megjelennek a tájékozódás végett. Egy szövegrész például így íródhat ki:

```
1: Ez az elso sor
2: ez pedig a masodik
3:*Ez az aktualis sor
4: Ez itt a 4. sor
```

(Az ékezetes betűket természetesen nem tudjuk beírni, de így olvashatóbb lesz a szöveg.)

A példában tehát az 1-től 4-ig terjedő sorszámok (valamint a kettőspontok és a csillag) nem részei a szövegnek, de általuk tudjuk meg, hogy az egyes sorszámokra milyen számmal lehet hivatkozni. A harmadik sor szövege előtt látható csillag (ne tévesszük össze az EDLIN prompt-jával) azt jelzi, hogy ez az ún. aktuális sor.

A parancsokban a sorokra általában sorszámokkal hivatkozhatunk (1-től 65529-ig), de két speciális jel is megadható a sorszámok helyett. Ezek jelentése a következő:

. (pont) az aktuális sort jelzi
(hashmark) az utolsó utáni sort jelzi

Abszolút sorszámok helyett az aktuális sorhoz viszonyított relatív hivatkozásokat is használhatunk. Ha egy parancsban +10-et (ill. -10-et) írunk, akkor ezzel az aktuális sornál 10-zel nagyobb (illetve 10-zel kisebb) sorszámra hivatkozunk.

A legtöbb EDLIN parancs egyetlen betűből áll, amely előtt és/vagy után paraméterek szerepelhetnek. A parancsokat kis vagy nagy betűvel is írhatjuk. Kivételt egyedül a "sor szerkesztése" parancs képez, amelynek nincs betűjele, csak az átszerkesztendő sor számából áll.

A numerikus paramétereket vesszővel vagy szóközzel választjuk el egymástól. Egy sorba több EDLIN parancsot is írhatunk, ilyenkor ezeket pontosvesszővel kell egymástól elválasztanunk. Ha a pontosvessző előtt egy "keresés" vagy "helyettesítés" parancs áll, akkor azok utolsó string-adatát <ENTER> helyett <F6>-tal kell lezárni. A parancsokat az EDLIN csak az <ENTER> billentyű lenyomásakor hajtja végre.

Ha a szövegbe a <Ctrl> billentyűvel képzett speciális karaktereket akarunk írni, akkor először a <Ctrl-V> billentyűket kell lenyomnunk, majd a kívánt karakter nagybetűs alakját. A <Ctrl-Z> kombinációt például így állíthatjuk elő:

<Ctrl-V> Z

Az EDLIN futtatása során a megszokott módon alkalmazhatjuk valamennyi DOS szerkesztő billentyűt (<F1>, <F2>, <F3>, <F4>, <F5>, <Del>, <Backspace>, <ESC>, kurzor jobbra és balra) a parancsok és a szövegek írásánál is. (E billentyűk szerepét a 2.4 fejezetben ismertettük.) A parancsok végrehajtását a <Ctrl>-<Break>

kombináció szakítja meg, és a <Ctrl-NumLock> <Shift-PrtSc>, <Ctrl-PrtSc> billentyűk is ugyanúgy működnek, mint a DOS-ban.

8.2.1 I (insert) - sorok beszúrása

Az I (insert, sorok beszúrása, beillesztése) parancs nélkülözhetetlen az új file-ok létrehozásánál, valamint a meglevő szövegek új sorokkal való bővítésénél is. A parancs hatására az EDLIN "beszúró üzemmódba" tér át, és az újonnan beírt sorokat a megadott számú sor elé illeszti be. A parancs általános formája:

[sorszám]I

Ha a sorszámot nem adjuk meg, akkor a beszúrás az aktuális (csillaggal jelzett) sor elé történik. Ha pedig a # jelet használjuk, vagy a létezőknél magasabb sorszámra hivatkozunk, akkor az új sorok az eddigi szöveg utolsó sora után kerülnek.

A beszúró üzemmódban tetszőleges számú új sort írhatunk be, az <ENTER>-rel lezárva. Az EDLIN mutatja a folyamatosan növekvő sorszámokat, amelyeket az új sorok kapnak. Ha nem akarunk több sort beszúrni, a <Ctrl-Break> billentyűkkel léphetünk ki a beszúró módból.

A beillesztések elvégzése után az utolsó beszúrt sort követő sor lesz az aktuális sor (akkor is, ha a szöveg végéhez toldottunk hozzá új sorokat). Az aktuális sort, valamint az utána álló sorokat (ha vannak ilyenek) az EDLIN újraszámozza, így helyreáll az egyesével folyamatosan növekvő számozás rendje.

Első példánkban egy új file létrehozásához használjuk fel a beszúró programot:

```
A>EDLIN B:UJ.TXT
New file
*I <ENTER>
  1:* Tevedni emberi dolog, de <ENTER>
  2:* igazan osszekutyulni valamit <ENTER>
  3:* csak szamitogeppel lehet. <ENTER>
  4:*<Ctrl-Break>
*
```

Az I parancsot sorszám nélkül adtuk ki, de mivel a memória eddig üres volt, ezért az új sorok (természetesen) az első sortól kezdve illesztődnek a file-ba. Három sort írtunk be <ENTER>-rel lezárva, majd a negyedik sor helyett lenyomtuk a <Ctrl-Break>-et. Erre az EDLIN kilépett a beszúró módból, és visszatért parancs-szintre, amint azt a prompt (csillag) megjelenéséből látjuk.

A fenti példában egészen rövid sorokat írtunk be. Lehetőségünk van azonban hosszú, de legfeljebb 254 karakteres sorok létrehozására is. Bár a 72. karakter begépelésekor eljutunk a képernyő jobb szélére, nyugodtan írhatjuk tovább a szöveget. Ilyenkor a sor folytatását az EDLIN a képernyő következő sorába írja. Egy logikai sor tehát - a megjelenítés szempontjából - több, maximum négy fizikai sorra tagolódhat. A 254. karakter beírása után az EDLIN hangjelzést ad, és több karaktert nem fogad el.

Ha a sorok írása közben hibát vétettünk, legegyszerűbben a <Backspace> (visszatörlés) billentyűvel, majd a hibás rész újbóli begépelésével javíthatunk. Felhasználhatjuk továbbá a DOS szerkesztőbillentyűit is (<F1>, <F3>, <Ins>, <Del> stb.).

Folytassuk a példát úgy, hogy az első sor elé illesztünk be újabb sorokat:

```
*1I
  1:*A megbizhatosag
  2:*otodik torvenye
  3:*<Ctrl-Break>
*L
  4:*Tevedni emberi dolog, de
  5: igazan osszekutyulni valamit
  6: csak szamitogeppel lehet.
*
```

Végül megnézzük, hogyan lehet beszúrni egy új sort az eddigi 3. sor elé:

```
*3I
  3:*(Murphy torvenykonyvebol):
  4:*<Ctrl-Break>
*L
  1: A megbizhatosag
  2: otodik torvenye
  3: (Murphy torvenykonyvebol):
  4:*Tevedni emberi dolog, de
  5: igazan osszekutyulni valamit
  6: csak szamitogeppel lehet.
*
```

Figyeljük meg a példánkban, mikor melyik lett az aktuális sor.

8.2.2 L (list) - sorok listázása

Az L (list, listázás) parancs hatására az EDLIN megjeleníti a kijelölt sorokat a képernyőn. Eközben az aktuális sorszám nem változik. A parancs általános formája:

[sorszám1][,sorszám2]L

Ha mindkét sorszámot megadtuk, akkor a sorszám1-től a sorszám2-ig terjedő sorok jelennek meg (beleértve a két határértéket is). A sorszám1 nem lehet nagyobb a sorszám2-nél, különben "Entry error" (hibás parancs) üzenetet kapunk.

Ha mindkét sorszámot elhagyjuk, és csak az L kulcsszót írjuk be, akkor általában az aktuális sor előtti tizenegy sor, az aktuális sor és az azt követő tizenegy sor (azaz összesen huszonhárom sor) jelenik meg a képernyőn. Kivétel ez alól, ha az aktuális sor előtt nincs tizenegy sor, vagyis az aktuális sor száma tizenkettőnél kisebb. Ilyenkor az EDLIN az első huszonhárom sort jeleníti meg.

Ha az első sorszámot hagyjuk el és a parancsot így írjuk:

,sorszám2 L

akkor a kiíratás az aktuális sor előtti tizenegyedik sorral kezdődik és a kijelölt sorral végződik. Kivétel, ha a megadott sorszám eleve kisebb, mint az aktuális sor előtti tizenegyedik sor száma. Ilyenkor az EDLIN úgy hajtja végre a parancsot, mintha mindkét sorszámot elhagytuk volna.

Ha csak az első sorszámot írjuk ki:

sorszám1 L

akkor a megadott sorszámtól kezdve összesen huszonhárom sor jelenik meg.

A példában feltételezzük, hogy a szövegben ötven sor szerepel, és mindegyikben utalás található a sor sorszámára. (Ez az 1.sor, Ez a 2.sor stb). Az aktuális sor legyen a negyvenötödik.

Nézzük meg ezek után néhány listázási parancs hatását:

```
*L
 34: Ez a 34. sor
 35: Ez a 35. sor
 36: Ez a 36. sor
   .
   .
   .
 45:*Ez a 45. sor
 46: Ez a 46. sor
 47: Ez a 47. sor
 48: Ez a 48. sor
 49: Ez a 49. sor
 50: Ez az 50. sor
*11,14L
 11:Ez a 11. sor
 12:Ez a 12. sor
 13:Ez a 13. sor
 14:Ez a 14. sor
*,38L
 34:Ez a 34. sor
 35:Ez a 35. sor
 36:Ez a 36. sor
 37:Ez a 37. sor
 38:Ez a 38. sor
*14L
 14:Ez a 14. sor
 15:Ez a 15. sor
 16:Ez a 16. sor
```

```
    .
    .
 35:Ez a 35. sor
 36:Ez a 36. sor
*
```

Nézzük át a példákat, és ellenőrizzük a megadott szabályok szerint, hogy miért pont ezeket a sorokat írta ki az EDLIN. (Az első listázási parancsnál nem lehetett az aktuális sor után tizenegy sort megjeleníteni, mert az 50. sornál vége lett a szövegnek.)

8.2.3 P (page) - listázás laponként

A P (page, listázás laponként) parancs nagyon hasonlít az előző pontban ismertetett L (listázás) parancshoz, ez is a megadott sorok képernyőn megjelenítését szolgálja. A fő különbség abban rejlik, hogy a P parancs (az L-lel szemben) megváltoztatja az aktuális sorszámot (az utolsóként kiírt sor lesz az aktuális sor).

Különbözik továbbá az első sorszámra vonatkozó alapértelmezés is. A parancs általános alakja:

[sorszám1][,sorszám2]P

Ha mindkét sorszámot megadtuk, akkor a sorszám1-től a sorszám2-ig terjedő sorok jelennek meg (beleértve a két határértéket is). A sorszám1 nem lehet nagyobb a sorszám2-nél, különben "Entry error" (hibás parancs) üzenetet kapunk.

Ha elhagyjuk az első sorszámot, akkor a kiíratás az aktuális sor utáni sorral fog kezdődni. Ha a második sorszámot hagyjuk el, akkor az elsőként megadott sortól kezdve összesen huszonhárom sor jelenik meg. Ha mindkét sorszámot elhagyjuk, és csak a P parancsot írjuk ki, akkor mindkét sorszámra az előbbiekben ismertetett alapértelmezés érvényesül.

A P parancs (paraméterek nélküli) ismételgetésével tehát kényelmesen végiglapozhatunk, végignézhetünk egy file-t az elejétől a végéig.

A példában most is feltételezzük, hogy a szövegben ötven sor szerepel és mindegyikben utalás található a sor számára (Ez az 1. sor, Ez a 2.sor stb). Az aktuális sor legyen a negyvenötödik. Jelenítsük meg a szöveget a képernyőn az első sortól az utolsóig:

```
*1P
 1: Ez az 1. sor
 2: Ez a 2. sor
 3: Ez a 3. sor
    .
    .
    .
 22: Ez a 22. sor
 23:*Ez a 23. sor
*P
 24: Ez a 24. sor
 25: Ez a 25. sor
```

```
      .
      .
      .
  45: Ez a 45. sor
  46:*Ez a 46. sor
*P
  47: Ez a 47. sor
  48: Ez a 48. sor
  49: Ez a 49. sor
  50:*Ez az50. sor
*
```

Nézzük meg a példában, hogyan változik az aktuális sorszám.

8.2.4 Sorszám - sor szerkesztése

Ha egy meglevő file-t szerkesztünk át, vagy az I parancs kiadása után magunk írunk be egy szöveget, szükség lehet a memóriában elhelyezkedő sorok módosítására, javítására. Ezt a célt szolgálja a - némileg rendhagyó - "sor szerkesztése" parancs, melynek általános formája:

[sorszám]

A parancsnak nincs betűjele, csak az átszerkeszteni kívánt sor számát kell megadnunk. Sorszám helyett a "." (pont) is használható, amely az aktuális sorra utal. Ha elhagyjuk a sorszámot és csak az <ENTER>-t nyomjuk meg, akkor az aktuális sor utáni sort módosíthatjuk. Utóbbi módszerrel kényelmesen lehet az egymás utáni sorokat javítani.

A "sor szerkesztése" parancs hatására a kijelölt sor megjelenik a képernyőn. Az alatta levő sorban ismét kiíródik a sor száma, ebben a sorban lehet a módosítást vagy újraírást végrehajtani. Az alábbi példában egy program szövegének 24. sorát módosítjuk:

```
*24
  24:*DCL SUM  FIXED BIN (15);
  24:*
```

Az utoljára megjelenített üres sorba tetszésünk szerinti új szöveget is beírhatunk, amely a korábbi tartalomtól független. Az <ENTER> lenyomása után ez lesz a sor új tartalma és ez a sor lesz az új aktuális sor. Példa a teljes újraírásra:

```
*24
  24:*DCL SUM FIXED BIN (15);
  24:*DECLARE (X,Y) FLOAT DEC (5);
*
```

Előfordulhat az is, hogy csak kisebb változtatásokra van szükség, ilyenkor a sor régi tartalmát - az írásmunka csökkentése érdekében - felhasználjuk, a módosításokat pedig a DOS szerkesztő billentyűivel (<F1>, <F2>, <F3>, <F4>, <F5>, <Ins>, <Del>,

<Backspace>, <ESC>, kurzor jobbra és balra) végezzük el. (E billentyűk szerepét a 2.4 fejezetben ismertettük.) A következő példában a SUM változó nevét S-re javítjuk ki:

```
*24
 24:*DCL SUM FIXED BIN(15);
 24:*DCL S
```

A szerkesztő parancs kiadása után ötször lenyomjuk az <F1>-et, erre megjelent a sor első öt karaktere. Ezután kétszer lenyomjuk a <Del> billentyűt, ami törli a következő két betűt ("UM"). Végül az <F3> gombbal kiíratjuk a sor hátralevő karaktereit:

```
*24
 24:*DCL SUM FIXED BIN (15);
 24:*DCL S FIXED BIN (15);
*
```

A sorba új szövegrészeket az <Ins> billentyű lenyomása után lehet beírni (<Ins> nélkül az újonnan begépelt karakterek a sor régi tartalmát módosítják, felülírják). A szerkesztő billentyűk használatára további példákat találunk a 2.4 fejezetben.

A változtatások végrehajtása után, a sor végén, nyomjuk le az <ENTER>-t, ekkor lép a módosított sor az eredeti helyébe. Az <ENTER>-t csak a sor végén szabad lenyomni, különben a sor hátralevő, nem látható része elvész. Ezután az utolsóként módosított sor lesz az aktuális sor.

A szerkesztő parancs kiadása után még meggondolhatjuk magunkat és változtatás nélkül hagyhatjuk a kijelölt sort (például, ha rájövünk, hogy nem is ezt a sort akartuk módosítani). A szerkesztésből- módosítás nélkül - a következő módszerekkel lehet kilépni:

- Lenyomjuk az <ENTER>-t, amikor a kurzor a sor első karakterén áll. (Vigyázat, ha a kurzor nem a sor elejére mutat, akkor a módosító sorban látható szöveg lép az eredeti sor helyébe!)
- A sor szerkesztése közben bármikor lenyomjuk a <Ctrl-Break> billentyűket.
- Ha módosítás nélkül léptünk ki a szerkesztésből, akkor az aktuális sor száma sem változik meg.

A szerkesztő paranccsal csak az egyes sorok tartalmát tudjuk módosítani, új sorok beillesztésére az "I", sorok törlésére a "D" parancsot kell használni.

Az összefoglaló példában egy levél 9. sorát módosítjuk. Az eredeti szöveg ez volt:

```
9: Kedves Zsuzsa, 5-kor varlak a mozi elott.
```

Hirtelen változás folytán a "Zsuzsa" nevet "Évá"-ra, a találkozó helyét pedig "mozi" helyett "diszkó"-ra kell módositanunk. A javítás lépései:

```
*9
 9:*Kedves Zsuzsa, 5-kor varlak a mozi elott
 9:*Kedves
```

A változatlanul hagyott első szót az <F2> billentyű, majd a nagy "Z" betű lenyomásával

ismételtük meg (az eredeti szöveg másolása a "Z" első előfordulásáig). Ugyanez lett volna a hatás az <F1> hétszeri lenyomásával is.

Ezután beírjuk az "Éva" nevet, és háromszor lenyomjuk a <Del> billentyűt (három betűvel hosszabb a "Zsuzsa" név). Ezt látjuk a képernyőn:

```
9:*Kedves Zsuzsa, 5-kor varlak a mozi elott.
9:*Kedves Eva
```

Nyomogassuk az <F1> billentyűt mindaddig, amíg a régi szöveget változtatás nélkül akarjuk átvenni. Tizenhét lenyomással tudjuk az ", 5-kor várlak a" szavakat megismételni:

```
9:*Kedves Zsuzsa, 5-kor varlak a mozi elott.
9:*Kedves Eva, 5-kor varlak a
```

Mivel a "diszkó" szó két betűvel hosszabb a "mozi"-nál, az első két karakter számára az <Ins> billentyűvel csinálunk helyet. Lenyomjuk az <Ins> gombot, majd a "d" és "i" betűket, végül az <Ins> újbóli lenyomásával lépünk ki a beszúró módból:

```
9:*Kedves Zsuzsa, 5-kor varlak a mozi elott
9:*Kedves Eva, 5-kor varlak a di
```

A "mozi" szó karakterei helyére egyszerűen begépeljük a "szkó" betűsorozatot, majd az <F3> lenyomásával átmásoltatjuk a sor hátralevő részét (az "előtt" szót). Ezzel befejeztük a változtatásokat és eljutottunk a sor végére. Most már lenyomhatjuk az <ENTER>-t:

```
9:*Kedves Zsuzsa, 5-kor varlak a mozi elott.
9:*Kedves Eva, 5-kor varlak a diszko elott.
*
```

8.2.5 D (delete)- sorok törlése

Most már tudjuk, hogyan kell új sorokat létrehozni és hogy lehet a meglevő sorokat módosítani. Ismerkedjünk meg ezek után a D (delete, törlés) paranccsal, melynek segítségével a feleslegessé vált sorokat teljes egészében kitörölhetjük.

A parancs általános formája:

[sorszám1][,sorszám2]D

Ha mindkét sorszámot megadjuk, akkor a sorszám1-től a sorszám2-ig terjedő sorok törlődnek (beleértve a két határértéket is.)

Ha elhagyjuk a második sorszámot, és a parancsot így írjuk:

sorszám1 D

akkor csak a kijelölt sor törlődik.

Ha csak a második sorszámot írjuk ki:

,sorszám2 D

akkor az aktuális sortól kezdve a megadott számú sorig terjedő szövegrészt törli az EDLIN.

Ha mindkét sorszámot elhagyjuk, és csak a D kulcsszót írjuk ki, akkor csak az aktuális sor törlődik.

A parancs végrehajtása után az utolsó kitörölt sort követő sor lesz az aktuális sor (akkor is ha az utolsó sort töröltük ki). Az aktuális sort, valamint az utánna álló sorokat az EDLIN újraszámozza, így helyreáll a folyamatos sorszámozás rendje.

A törlési parancsot a következő szövegen fogjuk gyakorolni:

```
1: Az EDLIN hibauzeneteinek listaja:
2:--------------------------------
3: Cannot edit .BAK file -rename file
4: (BAK file-t nem lehet szerkeszteni, nevezze at)
5: Disk full - write not completed
6: (A lemez betelt - a tarolas nem fejezodott be)
7: Entry error
8: (Hibas parancs)
9: Line too long
10: (Tul hosszu sor)
11: Must specify destination line number
12: (meg kell adni a cel-sor szamat)
13:*No room in directory for file
14: (Betelt a directory /tartalomjegyzek/ a lemezen)
15: Not enough room to merge the entire file
16: (Keves a memoria a teljes file beolvasasahoz)
17: Not found
18: (Nem talalhato)
```

Töröljük a 9-től 14-ig terjedő sorokat:

```
*9,14D
*L
 1: Az EDLIN hibauzeneteinek listaja:
 2:----------------------------
 3: Cannot edit .BAK file -rename file
 4: (BAK file-t nem lehet szerkeszteni, nevezze at)
 5: Disk full - write not completed
 6: (A lemez betelt - a tarolas nem fejezodott be)
 7; Entry error
 8: (Hibas parancs)
 9:*Not enough room to merge the entire file
 10: (Keves a memoria a teljes file beolvasasahoz)
 11: Not found
 12: (Nem talalhato)
*
```

A 15-18. sorokat újraszámozza az EDLIN (9-12. lett a számuk). A 9.sor lett az aktuális sor.

Ha most a 9-10. sorokat akarjuk törölni, a D parancsban elhagyhatjuk az első sorszámot, hiszen az alapértelmezés szerint a törlés az aktuális sortól kezdődik.

```
*10D
*L
 1: Az EDLIN hibauzeneteinek listaja:
 2: ------------------------------
 3: Cannot edit .BAK file -rename file
 4: (BAK file-t nem lehet szerkeszteni, nevezze at.)
 5: Disk full - write not completed
 6: (A lemez betelt - a tarolas nem fejezodott be)
 7: Entry error
 8: (Hibas parancs)
 9: *Not found
 10: (Nem talalhato)
*
```

Most is a 9. sor maradt aktuális, de a sor tartalma megváltozott.

Töröljük a harmadik sort:

```
*3D
*L
 1: Az EDLIN hibauzeneteinek listaja:
 2: ------------------------------
 3:*(BAK file-t nem lehet szerkeszteni, nevezze at)
 4: Disk full -write not completed
 5: (A lemez betelt -a tarolas nem fejezodott be
 6: Entry error
 7: (Hibas parancs)
 8: Not found
 9: (Nem talalhato)
*
```

Ismét a harmadik sort töröljük, de kihasználjuk, hogy ez az aktuális sor:

```
*D
*L
 1: Az EDLIN hibauzeneteinek listaja:
 2: ------------------------------
 3:*Disk full - write not completed
 4: (A lemez betelt - a tarolas nem fejezodott be)
 5: Entry error
 6: (Hibas parancs)
 7: Not found
 8: (Nem talalhato)
```

```
*
```

Legyünk óvatosak a D parancs alkalmazásakor, mert a törlés visszafordíthatatlan művelet. Biztonságosabb, ha mindig kiírjuk a törlendő sor(ok) számát, és nem hagyatkozunk az alapértelmezésekre.

8.2.6 E (end)- szerkesztés befejezése tárolással

A szöveges file létrehozásának vagy módosításának befejezésekor ki kell lépnünk az EDLIN programból. Ha a kialakított szöveget a lemezen meg akarjuk őrizni, akkor az E (end, vége) parancsot választjuk. Ha a szövegre a jelenlegi formájában nincs szükség, akkor a "Q"-val lépünk ki (lásd a következő pontban).

A "szerkesztés befejezése tárolással" parancs formája:

```
E
```

A parancs hatására a memóriában tárolt szöveg kiíródik a lemezen már korábban megnyitott .$$$ kiterjesztésű file-ba, majd - a sikeres tárolás után - a file neve megkapja az EDLIN indító parancsában előírt, végleges kiterjesztést (lásd a 8.1 pontban). Ha egy meglevő file-t dolgozunk fel, akkor a régi eredeti file is a lemezen marad, de nevének kiterjesztése .BAK-ra változik. Ha ugyanilyen nevű és .BAK végződésű file már volt a lemezen, akkor az most törlődik.

Ha az E parancs kiadásakor a lemezen nincs elegendő hely az új szöveg tárolására, akkor a szövegnek csak egy része kerül ki a lemezre, a memóriában levő szöveg pedig elvész. Ebben az esetben az eredeti file neve nem változik meg (nem lesz .BAK), a részlegesen letárolt új file pedig megtartja az ideiglenes .$$$ kiterjesztést.

Az EDLIN a file-ok végére kiteszi a file-vége jelet (<Ctrl>-Z).

Az E parancs a szöveg tárolása után visszatér a DOS parancs szintjére, amint az a példában is látszik:

```
*E
A>
```

8.2.7 Q (quit)- kilépés a szerkesztésből

Előfordulhat, hogy a memóriában elhelyezkedő szöveg végleges tárolására nincs szükségünk. Ennek néhány lehetséges oka:

- A lemezről beolvasott szöveget csak megnéztük, de nem módosítottuk. Ilyenkor felesleges a tárolás.
- Tévedésből másik file-t olvastunk be, nem azt, amelyiket módosítani akartuk.
- Hibás módosításokat, felesleges törléseket hajtottunk végre, ezért újra szeretnénk kezdeni az eredeti file feldolgozását.

A tárolás nélküli kilépés parancsának formája:

```
Q
```

A Q (quit, kilépés) parancs hatására az EDLIN törli a lemezről az új file számára megnyitott, .$$$ kiterjesztésű file-t. Ha a futást egy meglevő file beolvasásával kezdtük, akkor az eredeti file -a nevének megváltoztatása nélkül- a lemezen marad. A szövegen végrehajtott változtatások elvesznek, archív (.BAK) file nem keletkezik.

Mielőtt az EDLIN visszatérne a DOS-hoz, a biztonság kedvéért felteszi az "Abort edit (Y/N)?" (megszakítja a szerkesztést?) kérdést. Ha valóban tárolás nélkül akarjuk befejezni a futtatást, nyomjuk le az Y billentyűt (Y = yes, igen). Ha bármely más karakterrel válaszolunk, akkor az EDLIN nem tér vissza az operációs rendszerhez és folytathatjuk a szerkesztést, vagy kiadhatjuk az E parancsot.

Az alábbi példában Y-nal megerősítjük a kilépési szándékot:

```
*Q
Abort edit (Y/N)? Y
A>
```

8.2.8 M (move) - sorok áthelyezése

Ha a szövegben egy vagy több sort más helyre akarunk áthelyezni, eddigi ismereteink alapján a következőket tehetnénk: az I parancs kiadása után megfelelő helyre újból beírjuk az áthelyezni kívánt szöveget, majd a régi helyén feleslegessé vált sorokat a D parancscsal kitöröljük. Most megismerkedünk az M (move, áthelyezés) paranccsal, amely lényegesen leegyszerűsíti ezt a feladatot. A parancs általános formája:

[sorszám1],[sorszám2],sorszám3M

ahol

sorszám1 az áthelyezendő szöveg első sorát jelenti,
sorszám2 az áthelyezendő szöveg utolsó sorát jelenti
sorszám3 pedig azt jelenti, amelyik elé a kijelölt szöveget be kell illeszteni.

A harmadik sorszámot meg kell adni, de az első kettő elhagyható. Ha bármelyiket kihagyjuk, az aktuális sor száma lép a ki nem írt sorszám(ok) helyébe.

A szövegrészt önmagán belül nem lehet áthelyezni, ezért a sorszám3 nem eshet a sorszám1 és a sorszám2 közé, továbbá nem lehet egyenlő a sorszám2-vel sem, különben az "Entry error" (hibás parancs) üzenetet kapjuk.

A parancs végrehajtása után az EDLIN újraszámozza a sorokat, és az első áthelyezett sor lesz az aktuális sor.

A példánkban az alábbi szöveggel fogunk dolgozni:

```
1: ; Mintaprogram
2: RECEIVR PROC   FAR
3:         ASSUME CS:SUBSEG
4:         PUBLIC RECEIVR
5:         INT    10H
6:         POP    BX
```

```
 7:          PUSH    BX
 8:          MOV     DX,0
 9:*         MOV     BH,0
10:          MOV     AH,2
11: RECEIVR ENDP
```

Tegyük föl, hogy az ötödik és hatodik sor rossz helyen van és át kell őket tenni a tizenegyedik sor elé:

```
*5,6,11M
*L
      1: ; Mintaprogram
      2: RECEIVR PROC   FAR
      3:         ASSUME CS:SUBSEG
      4:         PUBLIC RECEIVR
      5:         PUS    BX
      6:         MOV    DX,0
      7:         MOV    BH,0
      8:         MOV    AH,2
      9:*        INT    10H
     10:         POP    BX
     11: RECEIVR  ENDP
*
```

Hogyan lehetne az első sort áthelyezni az utolsó sor után? Egy lehetséges megoldás:

```
*1,1,#M
*L
      1: RECEIVR PROC   FAR
      2:         ASSUME CS:SUBSEG
      3:         PUBLIC RECEIVR
      4:         PUSH   BX
      5:         MOV    DX,0
      6:         MOV    BH,0
      7:         MOV    AH,2
      8:         INT    10H
      9:         POP    BX
     10: RECEIVR ENDP
     11:*; Mintaprogram
*
```

A # (hashmark, kettőskereszt) az utolsó utáni sorra hivatkozás jele.

Most a tizenegyedik sor az aktuális sor.

Ha az M parancsban elhagyjuk az első két sorszámot, az áthelyezés az aktuális sorra fog vonatkozni:

```
*,,2M
```

```
*L
     1: RECEIVR PROC   FAR
     2:*;Mintaprogram
     3:         ASSUME CS:SUBSEG
     4:         PUBLIC RECEIVR
     5:         PUSH   BX
     6:         MOV    DX,0
     7:         MOV    BH,0
     8:         MOV    AH,2
     9:         INT    10H
    10:         POP    BX
    11: RECEIVR ENDP
*
```

8.2.9 C (copy)-sorok átmásolása

A C (copy, másolás) parancs hasonlít az előző pontban bemutatott M (áthelyezés) parancshoz, de a legfőbb különbség, hogy a C parancsban kijelölt sorok a helyükön maradnak, nem törlődnek a másolás után. Ezzel lehet tehát egy vagy több sort a szövegben megismételni. A parancs általános formája:

[sorszám1],[sorszám2],sorszám3[,számláló]C

ahol
sorszám1 az átmásolandó szöveg első sorát jelenti,
sorszám2 az átmásolandó szöveg utolsó sorát jelenti,
sorszám3 azt a sort jelzi, amelyik elé a kijelölt szöveget be kell másolni,
számláló pedig azt mutatja, hányszor kell a szöveget a másoláskor megismételni.

A paraméterek közül csak a harmadik sorszámot kötelező megadni. A ki nem írt paraméterekre a következő alapértelmezés érvényesül: a sorszám1 és a sorszám2 helyett az EDLIN az aktuális sor számát feltételezi. A számláló alapértelmezése: 1, tehát a szöveg egy példányban másolódik a kijelölt helyre.

A szövegrészt önmagán belül nem lehet átmásolni, ezért a sorszám3 nem eshet a sorszám1 és a sorszám2 közé, továbbá nem lehet egyenlő a sorszám2-vel sem, különben az "Entry error" (hibás parancs) üzenetet kapjuk.

A parancs végrehajtása után az EDLIN újra számozza a sorokat, és az első átmásolt sor lesz az aktuális sor.

A példánkban az alábbi szöveget fogjuk felhasználni:

```
1: Greer harmadik torvenye:
2: **************************
3: A szamitogepprogram
4:*nem kivansagaid, hanem
5: utasitasaid szerint
6: mukodik.
```

Duplázzuk meg a szöveg harmadiktól hatodikig terjedő sorait. A megismételt szövegrész közvetlenül a hatodik sor után kezdődjön:

```
*3,6,#C
*L
  1: Greer harmadik torvenye:
  2: **************************
  3: A szamitogepprogram
  4: nem a kivansagaid, hanem
  5: utasitasaid szerint
  6: mukodik.
  7:*A szamitogepprogram
  8: nem kivansagaid, hanem
  9: utasitasaid szerint
  10: mukodik.
*
```

Ismételjük meg az aktuális (hetedik) sort két példányban, a tizedik sor előtt:

```
*,,10,2C
*L
  1: Greer harmadik torvenye:
  2: **************************
  3: A szamitogepprogram
  4: nem kivansagaid, hanem
  5: utasitasaid szerint
  6: mukodik.
  7: A szamitogepprogram
  8: nem kivansagaid, hanem
  9: utasitasaid szerint
  10:*A szamitogepprogram
  11: A szamitogepprogram
  12: mukodik.
*
```

A példában kihasználtuk a sorszám1-re és a sorszám2-re vonatkozó alapértelmezéseket. Kiírtuk viszont a számláló paramétert, melynek hatására a kijelölt sor két példányban másolódott át.

8.2.10 S (search) - karaktersorozat keresése

Az S (search, keresés) parancs kiadásakor az EDLIN megkeresi a szövegben a megadott karaktersorozatot, és megjeleníti azokat a sorokat, amelyekben a keresett szövegrész előfordul. E parancs igen hasznosnak bizonyul, ha például egy programban az ALAPBER változó valamennyi előfordulását ellenőrizni akarjuk, vagy ha az Egri csillagok című regényben a Dobó Istvánra vonatkozó mondatokat szeretnénk kielemezni.

A parancs általános formája:

[sorszám1][,sorszám2][?]S[karaktersorozat]

ahol

sorszám1 jelöli azt a sort, amelynél a keresés kezdődik,
sorszám2 jelöli azt a sort, amelynél a keresés befejeződik,
? jelzi, hogy nem kell befejezni a keresést az első találat után,
karaktersorozat helyébe pedig a keresett szövegrészt kell írni.

A megadott karaktersorozat előfordulásait az EDLIN csak a sorszám1 és a sorszám2 által kijelölt tartományban keresi meg. A két sorszámot nem kötelező kiírni.

Ha a sorszám1-et elhagyjuk, akkor a keresés az aktuális sor utáni sortól kezdődik. Ha a sorszám2-t hagyjuk el, akkor a keresés a memóriában elhelyezkedő szöveg végéig fog tartani. Ha egyik sorszámot sem írjuk ki, akkor mindkét sorszámra az előbb leírt alapértelmezés érvényesül.

Ha a kérdőjelet nem adjuk meg, akkor az EDLIN megjeleníti az első olyan sort, amelyben szerepel a kijelölt karaktersorozat és ezzel be is fejezi a keresést. (Ha ilyen sort nem talált, akkor a "Not found" (nem található) üzenetet írja ki. Ha a parancsban szerepel a kérdőjel-paraméter, akkor az EDLIN az első találatot tartalmazó sor után felteszi az "O.K.?" kérdést és a keresést válaszunktól függően folytatja. Ha az Y (yes, igen) vagy az <ENTER> billentyűt nyomjuk le, akkor a keresés befejeződik. Bármely más válasz esetén az EDLIN tovább keres, és a következő olyan sort jeleníti meg, amelyben szintén szerepel a kijelölt szövegrész.

A kereséseket mindaddig folytathatjuk az "O.K.?" kérdésre adott nemleges válasszal, amíg a "Not found" üzenetet nem kapjuk.

A sikeres keresés után a megadott karaktersorozatot tartalmazó sor lesz az aktuális sor. Ha az EDLIN ilyet nem talál, akkor az aktuális sor nem változik.

A karaktersorozat beírásakor pontosan kell idéznünk a keresett szövegrészt; a nagybetűs ill. kisbetűs írásmód ebből a szempontból különbséget jelent. (Ha az "EDLIN" szót keressük, a gép nem jelez találatot az "edlin" szónál.) A karaktersorozatot közvetlenül az S betű után kell írni, mert az esetleges közbenső szóközöket az EDLIN a karaktersorozat részének tekinti. A karaktersorozat végét általában az <ENTER> lenyomásával jelezzük (kivéve, ha az S paranccsal egy sorban további utasításokat akarunk írni, ilyenkor az <F6> vagy a <Ctrl-Z> billentyűket használjuk az <ENTER> helyett).

Ha a parancsban nem adjuk meg a keresendő karaktersorozatot, akkor az EDLIN a legutóbbi S, ill. R (replace, helyettesítés) parancsban szereplő szövegrész előfordulásait fogja keresni. Ez az ismétlési eljárás nagyon hasznos, mert sok írásmunkát takaríthat meg. Az S parancs hatására kijelzett sorokat az EDLIN aktuálissá teszi, így ezeket a "."(pont) utasítással lehet átszerkeszteni.

A keresési parancs illusztrálására a következő szöveget fogjuk felhasználni:

```
1: Cahm axiomaja
2: ********************
```

```
 3: (az EDLIN-re is ervenyes)
 4:*Ha mar minden kiserlet
 5: csodot mond, olvasd el
 6: a hasznalati utasitast
 7:
 8: Sattinger torvenye
 9: **********
10: (a szamitogepre is vonatkozik)
11: Jobban mukodik, ha
12: bedugod a konnektorba.
```

Keressük meg az "a" névelő előfordulásait. A keresett karaktersorozat ne az "a" betű legyen, mert ez a szavak belsejében található "a" betűket is kiválasztaná. Ezért az "a" után egy szóközt hagyunk ki, így csak azokat az "a"-kat kapjuk meg, amelyek a szavak végén állnak.

```
*1?Sa <Enter>
 4:*Ha mar minden kiserlet
O.K.? N
 6: a hasznalati utasatast!
O.K.? N
10: (a szamitogepekre is vonatkozik)
O.K.? N
12: bedugod a konektorba.
O.K.? N
Not found
*
```

A szövegben előforduló valamennyi névelőre szükségünk volt, azért előírtuk, hogy a keresés az első sornál kezdődjön. A második sorszámot elhagytuk, így a keresés a szöveg végéig tartott. A kérdőjelet azért adtuk meg az S parancsban, hogy a keresés ne álljon le az első találat után.

A negyedik sorban a "Ha" szót találta meg a gép, az első és a tizenegyedik sorban viszont azért nem jelzett találatot a szóvégi "a" betűknél, mert azok után nem található szóközjel, csak a sor végét jelző kocsivissza-karakter.

Az EDLIN "O.K.?" kérdésére mindig nemleges választ adtunk, hogy folytatódjon a keresés. Amikor az EDLIN a szöveg végére ért és további egyezéseket nem talált, kiírta a "Not found" üzenetet. A fenti műveletek elvégzése után a negyedik maradt az aktuális sor, mert az utolsó keresés sikertelen volt, a "Not found" üzenettel végződött.

A következő példában keressük meg a második cím aláhúzását (a csillagokból álló sort), és helyettesítsük kötőjelekkel. A parancs kiadásakor emlékszünk arra, hogy ez a sor a negyedik és a tizedik sor között helyezkedik el:

```
*4,10S***
 9:**********
*
```

```
 9:******************
 9:------------------
*L
 1: Cahn axiomaja
 2: ******************
 3: (az EDLIN-re is ervenyes)
 4: Ha mar minden kiserlet
 5: csodot mond, olvasd el
 6: a hasznalati utasitast!
 7:
 8: Sattinger torvenye
 9:*------------------
 10: (szamitogopre is vonatkozik)
 11: Jobban mukodik, ha
 12: bedugod a konnektorba.
*
```

A szerkesztő parancsban a "." (pont) alkalmazásakor kihasználtuk, hogy az utoljára megtalált sor lett az aktuális sor. A sormódosítást úgy végeztük el, hogy a régi tartalom helyett tizennyolc kötőjelet írtunk be. Az átszerkesztés után is a kilencedik sor maradt az aktuális.

8.2.11 R (replace) - karaktersorozat helyettesítése

Az R (replace, helyettesítés) parancs végrehajtásának első fázisa, a keresés, megegyezik az S parancsnál leírtakkal. A második fázisban viszont az R parancs módosítja a megtalált karaktersorozatot; helyettesíti az általunk megadott szöveggel. Egy gyakorlati példa; az R paranccsal egy lépésben módosíthatjuk a programunkban a CIM változó valamennyi előfordulásánál a változó nevét LAKCÍM-re. A parancs általános formája :

[sorszám1][,sorszám2][?]R[karaktersor][<F6>karaktersor2]

ahol

sorszám1 jelöli azt a sort, amelynél a keresés és helyettesítés kezdődik,
sorszám2 jelöli azt a sort, amelynél a keresés és helyettesítés befejeződik,
? megadásakor a helyettesítések nem automatikusan mennek végbe, hanem az EDLIN minden találatnál megkérdezi, hogy elvégezze-e a helyettesítést,
karaktersor1 helyébe a keresett szövegrészt kell beírni,
karaktersor2 helyébe pedig a keresett szövegrész helyére írandó új szöveget kell írni.

A megadott karaktersorozat előfordulásait az EDLIN csak a sorszám1 és a sorszám2 által kijelölt tartományban keresi meg. A két sorszámot nem kötelező kiírni. Ha a sorszám1-et elhagyjuk, akkor a keresés és helyettesítés az aktuális sor utáni sortól kezdődik. Ha a sorszám2-t hagyjuk ki, akkor a keresés és helyettesítés a memóriában elhelyezkedő szöveg végéig fog tartani. Ha egyik sorszámot sem írjuk ki, akkor mindkét sorszámra az előbb leírt alapértelmezés érvényesül.

Ha a kérdőjelet nem adjuk meg, akkor az EDLIN - a kijelölt tartományban - a karaktersor1 valamennyi előfordulását kicseréli a karaktersor2-ben megadott szövegre. A módosított sorok eközben megjelennek a képernyőn. Ha az EDLIN egy módosítandó szöveget sem talál, akkor a "Not found" (nem található) üzenetet írja ki.

Ha a parancsban szerepel a kérdőjel-paraméter, akkor az EDLIN a karaktersor1 minden helyettesítése után leáll, és felteszi az "O.K.?" kérdést. Ha a helyettesítéssel egyetértünk, akkor nyomjuk le az Y (YES) vagy az <ENTER> billentyűt.Bármely más válasz esetén az EDLIN visszaállítja a karaktersor1-ben megadott eredeti tartalmat. Mindkét esetben a keresés és helyettesítés tovább folytatódik a kijelölt tartomány végéig.

A helyettesítések végrehajtása után az utolsó módosított sort megelőző sor lesz az aktuális sor.

A karaktersor1 és a karaktersor2 beírásakor nagyon pontosan kell idézni a keresett és helyettesítő szövegeket. Ne felejtsük el a kis- és nagybetűs írásmód közötti különbséget. A karaktersor1-et közvetlenül az R betű után, a karaktersor2-t pedig az <F6> után kell írni, mert az esetleges közbenső szóközöket az EDLIN a karaktersorozatok részének tekinti. A karaktersor1 végét az <F6> vagy a <Ctrl>-Z billentyűkkel kell jelezni (vagy az <ENTER>-rel, ha a karaktersor2 nem szerepel a parancsban). A karaktersor2 végén általában az <ENTER>-t használjuk (kivéve ha az R paranccsal egy sorban további utasításokat akarunk írni, ilyenkor az <F6> vagy a <Ctrl>-Z billentyűket nyomjuk le az <ENTER> helyett).

Ha a parancsban az <F6> billentyű lenyomása után nem írunk be helyettesítő szöveget (a karaktersor2-t), akkor az EDLIN törli a karaktersor1 valamennyi előfordulását a kijelölt tartományban. Ha az <F6>-ot sem adjuk meg, akkor a gép a legutóbbi R parancsban szereplő helyettesítő szöveget fogja felhasználni. Ha mindkét karaktersorozatot elhagyjuk, akkor a keresési szöveg (karaktersor1) helyébe a legutóbbi S, ill. R parancsban szereplő szövegrész lép, a helyettesítő szöveg (karaktersor2) helyett pedig az EDLIN a legutóbbi R parancs helyettesítő szövegét használja föl (ha volt ilyen).

A helyettesítési parancs alkalmazását a következő szövegen mutatjuk be:

```
1:          IF KERESET > 15300
2:             MOVE 0.15 TO NYUGJAR
3:          MULTIPLY KERESET BY NY.
4:          MOVE NY TO NYUGJAR.
5:          SUBTRACT NYUGJAR FROM KERESET
6:*            GIVING KIFIZ.
```

A példában egyszer hibásan a NYUGJAR változót írtuk NY változó helyett (a második sorban). Automatikus helyettesítést azonban nem végezhetünk, mert a NYUGJAR többi előfordulását nem kell megváltoztatni:

```
*1?RNYUGJAR<F6>NY
 2:           MOVE 0.15 TO NY
*O.K.? Y
 4:        MOVE NY TO NY.
*O.K.? N
```

```
 5:        SUBTRACT  NY FROM KERESET
*O.K.? N
*L
 1:*       IF KERESET > 153000
 2:           MOVE 0.15 TO NY.
 3:        MULTIPLY KERESET BY NY.
 4:        MOVE NY TO NYUGJAR.
 5:        SUBTRACT NYUGJAR FROM KERESET
 6:           GIVING KIFIZ.
*
```

Az R parancsban elhagytuk a második sorszámot, ezért a végrehajtás az utolsó sornál fejeződött be. Az "O.K.?" kérdésekre adott nemleges válaszok esetén az EDLIN visszaállította a sorok eredeti tartalmát. Módosítsuk végül a teljes szövegben a KERESET változó nevét JOVEDELEM-re. Az automatikus helyettesítés módszerét alkalmazhatjuk;

```
*1RKERESET<F6>JOVEDELEM
 1:*          IF JOVEDELEM > 15300
 3:           MULTIPLY  JOVEDELEM BY NY.
 5:           SUBTRACT NYUGJAR FROM JOVEDELEM
*L
 1:           IF JOVEDELEM > 15300
 2:              MOVE 0.15 TO NY.
 3:           MULTIPLY JOVEDELEM BY NY.
 4:*          MOVE NY TO NYUGJAR.
 5:           SUBTRACT NYUGJAR FROM JOVEDELEM
 6:              GIVING KIFIZ.
*
```

8.2.12 T (transfer) - sorok beszúrása lemezről

A T (transfer, beszúrás, beolvasás lemezről) parancs segítségével egy lemezes adatállomány tartalmát lehet a memóriában elhelyezkedő sorok közé beilleszteni. Ilyen módon több file szövegét tudjuk egybeszerkeszteni. A parancs általános formája:

[sorszám]T[d:]file-név[.kit]

ahol

sorszám jelöli azt a sort, amelyik elé a file szövegét be kell illeszteni,
[d:]file-név[.kit] pedig a beolvasandó file specifikációja.

A sorszámot nem kötelező kiírni. Ha elhagyjuk, akkor a beillesztés az aktuális sor elé történik. A beolvasandó file specifikációjában nem lehet utat (path) megadni; a file-t az EDLIN mindig az adott egység aktuális könyvtárában keresi.

A művelet végrehajtása után az EDLIN újraszámozza a sorokat. Az aktuális sor az első beolvasott sor lesz.

Ha a file beolvasása közben megtelik a memória, akkor a "Not enough room to merge the entire file" (kevés a memória a teljes file beolvasásához) hibaüzenetet kapjuk.

A példánkban a következő szöveggel indulunk:

```
1:          READ (*,6001) IE,IV
2: 6001     FORMAT (2I3)
3:          CALL GYOK (IE,IV)
4:*         END
```

Tegyük fel, hogy az aktuális meghajtón az SQRT nevű állományban helyezkedik el az a szubrutin, amelyet a fenti program végéhez szeretnénk csatolni. Ehhez a # jelet használjuk fel, amely az utolsó utáni sor szimbóluma:

```
*#TSQRT
*L
 1:        READ (*,6001)IE,IV
 2: 6001   FORMAT (2I3)
 3:        CALL GYOK (IE,IV)
 4:        END
 5:*       SUBROUTINE GYOK (K,L)
 6:        WRITE (*,6000) (I,I=0,9),
 7:       X  (I,(SQRT(I+0.1*J),J=0,9),I=K,L)
 8; 6000  FORMAT (3H  ,10(6H  0.,I1
 9:       X  /,10(/,2HO ,I2,1X,10(1X,F6.3)))
 10:        END
*
```

A beolvasott sorokat az EDLIN a memóriában elhelyezkedő sorok után illesztette.

8.2.13 W (write) - sorok kiírása lemezre

A W (write, kiírás lemezre) parancsot akkor használjuk, ha a feldolgozandó file nem fér be a számítógép memóriájába és a szöveg létrehozását, módosítását több részletben kell elvégeznünk. Egyébként a "W" használata nélkül, az E paranccsal írhatjuk ki a teljes file-t a lemezre.

Ha az EDLIN indító parancsában megnevezett file már eleve túl hosszú volt, akkor az EDLIN csak annyi sort olvas be, amennyi a szabad memória 75%-át tölti ki. A memóriában elhelyezkedő szöveget (vagy annak egy részét) ki kell íratni a lemezre, ha

- a memória teljesen megtelt, és/vagy
- a szöveg további, még a lemezen elhelyezkedő részeit akarjuk a memóriába beolvasni és feldolgozni.

A parancs általános formája:

[n]W

ahol az n helyébe egy pozitív egész számot írhatunk.

A W parancs hatására az EDLIN kiírja a lemezre (a .$$$ kiterjesztésű állományba) a memóriában elhelyezkedő szöveg első n sorát. Ha nem adtuk meg az n paramétert, akkor az EDLIN addig írja a sorokat a lemezre, ameddig a memória telítettsége 25%-ra nem csökken. (Ha a szöveg a memóriának kevesebb mint a 25%-át foglalta el, akkor a paraméter nélküli W parancs hatástalan marad.)

A lemezre kiírt sorok törlődnek a memóriából, további feldolgozásuk csak az EDLIN újraindítása után lehetséges. A memóriában maradt sorok egytől kezdve újraszámozódnak. Az aktuális sor az új első sor lesz.

A memória felszabadítása után beolvashatjuk (és feldolgozhatjuk) az állomány következő részét. Erre a célra az A parancs szolgál, melyet a következő pontban ismertetünk.

Ha kíváncsiak vagyunk, hány sor található a memóriában, adjuk ki a

#I

parancsot. Ekkor megjelenik az utolsó sor számánál eggyel nagyobb sorszám. Nem akarunk újabb sorokat beírni, ezért a beszúró módból mindjárt ki is lépünk a <Ctrl>-<Break>-kel.

Tegyük fel, hogy memóriában 527 sor helyezkedik el, de a szöveg vége még a lemezen található. A további sorok beolvasása előtt helyet kell csinálnunk a tárban.

Néhány példa erre a műveletre:

*W

A paraméter nélkül kiadott parancs felszabadította a memória 75%-át

*600W

A létező sorok számánál nagyobb értéket adtunk meg. Az EDLIN az 527 sort kiírta a lemezre, a memória jelenleg üres.

*300W

Az 1-300. sorok kerültek ki a lemezre, az eddigi 301. sor lett az új első (és aktuális) sor. A tárban hely szabadult fel a további sorok beolvasására.

8.2.14 A (append) - sorok hozzáfűzése lemezről

Az A (append, sorok hozzáfűzése) parancsot az előző pontban leírt W paranccsal együtt használjuk, ha a módosítandó szöveg túl hosszú ,és feldolgozását csak részletenként lehet elvégezni. Az A parancs - feltéve, hogy elegendő hely van a gép memóriájában - a lemezes adatállomány következő részét beolvassa és hozzáilleszti a tárban elhelyezkedő szöveg végéhez. A parancs általános formája:

[n]A

ahol n helyébe egy pozitív egész számot írhatunk.

Ha megadtuk az n paramétert, akkor az EDLIN az indító parancsában megnevezett lemezes adatállományból beolvassa a következő n darab sort. Ha a paramétert nem írtuk ki,

akkor az EDLIN addig olvassa be a sorokat, ameddig a memória eléri a 75%-os telítettséget. Amennyiben a telítettség 75%-on felül volt, nem történik beolvasás.

Az újonnan beolvasott sorok mindig a tárban elhelyezkedő sorok után íródnak (sorszámaikat is ennek megfelelően kapják). Az aktuális sorszám nem változik.

Az EDLIN az "End of input file" (input file vége) üzenettel jelzi, amikor a feldolgozandó szöveg utolsó sorait is beolvasta a memóriába.

Az A parancs végrehajtása után elvégezhetjük a szöveg következő részének módosítását, átszerkesztését. A módosítás után a memóriában található sorokat a W paranccsal vihetjük ki a lemezre, hogy helyünk legyen a szöveg következő sorainak beolvasására.

Nézzük meg az alábbi példát:

```
A>EDLIN HOSSZU.TXT
*
```

Nem jelent meg az "End of input file" üzenet, tehát a megnevezett szöveg nem fér el egyszerre a memóriában. Vigyük ki lemezre az első 200 sorát, és olvassuk be a következő 200-at:

```
*200W
*200A
*
```

Most a memóriában az eredeti szöveg 201-től kezdődő sorai helyezkednek el (természetesen 1-től számozva). A szöveg ellenőrzése, módosítása után olvassuk be az újabb sorokat:

```
*W
*A
End of input file
*
```

A W parancs annyi sort írt ki a lemezre, amennyi a memória telítettségét 25%-ra csökkentette. A paraméter nélküli A parancs a tárat 75%-ra töltötte volna fel, de közben véget ért az input szöveg. Most tehát az eredeti szöveg utolsó sorai találhatók a memóriában.

9. A nyomkövető-hibakereső program (DEBUG)

Az alábbiakban bemutatott DEBUG rendszerprogram elsősorban a gyakorlott programozók számára készült, akik jól ismerik a számítógép felépítését és mikroprocesszorának működésmódját. A kevesebb tapasztalattal rendelkező felhasználóknak is javasoljuk, hogy tekintsék át, milyen feladatokra alkalmas a DEBUG. Az egyszerűbb parancsokat ők is bátran kiadhatják.

A DEBUG nyomkövető-hibakereső program lehetővé teszi, hogy a számítógép regisztereiben, operatív tárában (memóriájában) és lemezes háttértárolóiban található (gépi kódú) programok tesztelésére és javítására használjuk. A DEBUG futtatásakor a következő alapvető funkciók állnak rendelkezésünkre:

- a mikroprocesszor regisztereinek kiíratása és módosítása,
- az operatív tár tartalmának megjelenítése és módosítása,
- assembly nyelvű utasítások írása és az operatív tárba való elhelyezése,
- a memóriában található program visszafordítása assembly nyelvre (disassemblálás),
- az operatív tárban elhelyezkedő gépi kódú program futtatása (esetleg a kijelölt pontokon való megállással vagy lépésenkénti végrehajtással),
- adott tartalom keresése a memóriában,
- a memória kijelölt területének feltöltése a megadott tartalommal,
- a memória két területének összehasonlítása,
- az operatív tár kijelölt területének átmásolása,
- lemezen elhelyezkedő file-ok, ill. kijelölt lemezes szektorok betöltése a memóriába,
- a memória adott területének kiírása egy lemezes file-ba vagy kijelölt lemezes szektorokba,
- beolvasás a kijelölt portról (input-output csatornáról) és az adat megjelenítése,
- adott byte kivitele a kijelölt portra,
- hexadecimális összeadás és kivonás elvégzése.

A felsorolt műveletek segítségével a gépi kódú (assembly nyelven írt, illetve magasszintű nyelvről lefordított) programok kipróbálása a következő lépésekben történhet. Először betöltjük a programot a lemezről a memóriába. Ezután a kritikus pontokra helyezett megszakító utasításokkal (vagy lépésenkénti végrehajtással) lefuttatjuk a programot, és a megszakítások elérésekor megvizsgáljuk azt az információt, amelyet a DEBUG kiír a programvégrehajtás állásáról (regiszterek, fontosabb memóriaterületek tartalmát stb.). Szükség esetén a DEBUG mini assemblerével javításokat is végezhetünk a programon.

A számítógép 8088-as (vagy azzal kompatibilis) mikroprocesszorának utasításait nem ismerő programozók is végrehajthatnak kisebb módosításokat a programokon. Ha megtaláljuk az idegen nyelven kommunikáló programokban a futáskor megjelenített szövegeket, könnyen átírhatjuk azokat magyarra. A másik hasznos módosítás: ha egy program csak az "A" betűjelű floppy egységről hajlandó egy file-t beolvasni, mi pedig a "C" jelű fix lemezt akarjuk használni, keressük meg a programban a file-ra való hivatkozást (például: A:HELP.TXT), majd írjuk át az A betűt C-re.

A teljesség kedvéért megjegyezzük, hogy a DEBUG-nál nagyobb tudású és kényelmesebb kezelésű nyomkövető-hibakereső programok is forgalomban vannak. Utóbbiak néhány

többletszolgáltatása: teljes képernyős szerkesztés, memóriacímekre szimbólikus nevekkel való hivatkozási lehetőség stb.

A DEBUG működésének teljes megértéséhez szükséges a mikroprocesszor regisztereinek és utasításkészletének, valamint a számítógép felépítésének ismerete. Az irodalomjegyzékben felsorolunk olyan könyveket, amelyek ezekkel a kérdésekkel részletesen foglalkoznak.

9.1 A DEBUG indítása

A DEBUG program a DOS rendszerlemezen található, DEBUG.COM néven, de szükség esetén bármely más lemezre átmásolhatjuk. Ha a program az aktuális lemezegység aktuális alkönyvtárában (vagy a PATH parancsban kijelölt úton) elérhető, akkor a következő paranccsal indíthatjuk el:

DEBUG [[*d:*][*út*]*file-név*[*.kit*] [*par1*] [*par2*]]

ahol

[*d:*][*út*]*file-név*[*.kit*]

a megvizsgálni kívánt adat- vagy program-file specifikációja,

par1 és *par2*

pedig a megnevezett program futtatásához szükséges paraméterek.

A jelölésekből látszik, hogy a file-specifikáció és a paraméterek kiírása sem kötelező. Ha a file-specifikációt megadjuk az indító parancsban, akkor a DEBUG első dolga lesz, hogy betölti a memóriába a megnevezett file-t. Ha a tesztelni kívánt program paramétereket vár a parancs-sorban, akkor azokat a szokott módon a program neve után kell írni. A BASIC interpretert például a

BASIC

paranccsal indíthatjuk. A

BASIC RAKETA

parancs hatására pedig a BASIC interpreter az indulása után mindjárt betölti a RAKETA.BAS programot, és el is kezdi futtatni. Ha viszont nemcsak futtatni akarjuk a programot, hanem a DEBUG-gal nyomon is szeretnénk követni, hogyan futtatja a BASIC a RAKETA programot, akkor a következő indító parancsot kell kiadnunk:

DEBUG BASIC.COM RAKETA

Az alábbi példában az A jelű floppy SZOVEG nevű alkönyvtárából töltjük be a LEVEL.DOC nevű file-t (a DEBUG program a C lemezen helyezkedik el):

C:DEBUG A:\SZOVEG\LEVEL.DOC

Ha a file-specifikációt nem adjuk meg, akkor a DEBUG nem tölt be semmit a memóriába, így csak a tárba korábban bekerült adatokat vizsgálhatjuk, módosíthatjuk. Az N (na-

me) és az L (load) parancsok segítségével azonban még lehetőségünk lesz file-ok betöltésére.

A DEBUG indulásakor a regiszterek és a flag-ek (jelzők) a következő értékeket kapják:

- A négy szegmensregiszter (CS, DS, ES és SS) a szabad memóriaterület aljára mutatnak, azaz a DEBUG program fölötti első szegmensre.
- Az utasításmutató (IP) értéke hexadecimális 0100.
- A veremmutató (SP) vagy a szegmens végére mutat, vagy a COMMAND.COM tranziens részének aljára (amelyik cím alacsonyabb). A 6-os relatív címen (offset-en) található szegmensméret-mutatót a DEBUG hexadecimális 100-zal csökkenti, mert ekkora területet kap a verem.
- A további regiszterek (AX, BX, CX, DX, BP, SI, DI) a nulla értéket kapják, kivéve, ha az indító parancsban egy file betöltését is előírtuk. Utóbbi esetben a BX és a CX regiszterek a betöltött file hosszát adják meg, byte-okban kifejezve (a hossz alacsony helyiértékű része a CX, magasabb helyiértékű része a BX regiszterbe kerül).
- A flag-ek (jelzőbitek) törlődnek, értékük nulla lesz, kivéve a megszakítás- (interrupt) flag-et, amely 1-re állítódik (megszakítás engedélyezve).
- A lemez-puffer (Disk Transfer Area) kezdőcíme a kódszegmensben hexadecimális 80 lesz.

További megjegyzések a betöltött file-lal kapcsolatban:

A betöltött program futása során nem igényelhet további memóriát, mivel a DEBUG már valamennyi szabad területet lefoglalta a program számára.

Ha a betöltött program a .EXE kiterjesztéssel rendelkezik, akkor a DEBUG elvégzi a futtatáshoz szükséges előkészítő tevékenységeket. Végrehajtja az előírt relokációt (a program áthelyezhetőségéből adódó címmódosításokat) és beállítja a file-ban megadott értékekre a szegmensregisztereket, a veremmutatót és az utasításmutatót. A DS és az ES regiszterek azonban a legalacsonyabb című szegmensben elhelyezkedő ún. programszegmens-prefixre (PSP) mutatnak, a BX és a CX regiszterek pedig a program méretét tartalmazzák (ez kisebb, mint a file mérete).

Ha a programszerkesztőnek (LINK) megadtuk a /H paramétert, akkor a DEBUG a programot a memória lehető legmagasabb című részére helyezi el.

Ha a betöltött file .HEX kiterjesztést visel, akkor a file-ban ASCII-kódolású hexadecimális számjegyeknek kell szerepelniük (egy byte-on egy hexadecimális számjegy, karakteres ábrázolással). A DEBUG a hexadecimális jegyeket a betöltés során binárissá alakítja át.

9.2 A DEBUG parancsai

A DEBUG elindítása után egy kötőjel (-) jelzi a képernyőn, hogy a program a parancsainkat várja. A kötőjelet a DEBUG felszólító jelének, prompt-jának szokták nevezni. A kötőjel mellé kell beírnunk a DEBUG speciális parancsait.

A DEBUG parancsai egy betűből állnak, de a parancsokat általában további paraméterek követik. A parancsok beírásakor a kis- és a nagybetűket is használhatjuk. A parancsok végrehajtása csak akkor kezdődik el, amikor az < ENTER > billentyűt lenyomjuk.

A parancsokat és a paramétereket határoló jelekkel lehet egymástól elválasztani. Az elválasztás azonban csak két egymást követő hexadecimális szám között kötelező. Határoló jelnek szóközt vagy vesszőt használjunk.

A <Ctrl-Break> billentyűk együttes lenyomása félbeszakítja az éppen végrehajtás alatt álló parancsot.

A DEBUG futtatásakor szabadon használhatjuk a DOS szerkesztő billentyűit (ezeket a 2.4 pontban ismertettük).

Ha hibásan írtunk be egy DEBUG parancsot, akkor a program az "error" (hiba) szöveget jeleníti meg, és az első, értelmetlennek talált karakterre mutat. Ilyenkor a parancsot - hibátlanul - újra be kell írnunk.

A legtöbb parancsban paraméterként tárcímeket kell megadnunk. A címek írásmódját az alábbi általános forma mutatja:

[*szegmens*:]*relcím*

ahol

szegmens egy (tizenhat byte-os határon kezdődő) társzegmens kezdőcímét adja meg,
relcím pedig a szegmens kezdőcímétől számított relatív cím (offset).

Az ilyen módon felírt címből a tényleges (abszolút) címet úgy számíthatjuk ki, hogy a szegmenscímet megszorozzuk tizenhattal, majd hozzáadjuk a relatív címet.

A szegmens helyére nemcsak egy konkrét címet, hanem egy szegmensregiszter nevét (CS, DS, ES, SS) is beírhatjuk. Ebben az esetben a szegmens kezdőcímét a kijelölt regiszter tartalma határozza meg. Ha a szegmenscímet nem adjuk meg (ilyenkor a kettőspontra sincs szükség), akkor az egyes parancsoknál leírt alapértelmezés érvényesül.

A szegmens- és a relatív címet maximum négyjegyű hexadecimális számok formájában kell beírni. Vigyázzunk, hogy csak fizikailag létező tárcímekre hivatkozzunk, különben értelmetlen eredményt kapunk.

Néhány példa címek megadására:

1A3

Ez a parancstól függő alapértelmezés szerinti szegmensen belül az 1A3 relatív címet jelenti. (Az alapértelmezéseket az egyes parancsok leírásánál ismertetni fogjuk.)

12C8:A236

Az abszolút cím kiszámítása:

```
  12C80
+ A236
------
 1CEB6
```

CS:56

Ez a CS regiszter tartalma által meghatározott szegmensen belüli 56-os relatív címre utal.

Egyes parancsokban nem egyetlen tárcímet kell megadnunk, hanem a memóriacímek egy tartományát, intervallumát. Az intervallum egyik lehetséges felírása (általános alakban):

cím cím

ahol az első címet a fent ismertetett valamelyik formában kell megadni, a második cím pedig csak egy relatív címből állhat. A két címet szóköz vagy vessző választja el. Az intervallum ennél a felírásnál az elsőtől a második címig terjed (beleértve a határokat is). Példák memória-intervallumok megadására:

CS:1000 1F00
2000:1234,1235
F000 FFFF

Az intervallumok megadásának második formájában a kezdőcím mellett az intervallum hossza szerepel (byte-ban kifejezve):

cím L hossz

ahol a hosszat hexadecimálisan kell megadni. A hossz értékének maximuma hexadecimális 10000 lehet, de a négyjegyű ábrázolási korlát miatt az 10000 érték helyett 0-t kell írni. Néhány példa:

ES:100 L 2F
A000 L 0

Számos parancs szintaktikus leírásánál találkozunk majd az adatlista fogalmával. Az adatlista szóközzel vagy vesszővel elválasztott byte- és string-értékekből áll. A byte-okat egy- vagy kétjegyű hexadecimális számmal ábrázolhatjuk. A stringértékek idézőjelek vagy aposztrófok közé zárt karakterek lehetnek. Példák az adatlistára:

5C "xyz" 4 11 'abcd'
'ez egy "string"',FF,0

A stringeknél az egyes byte-ok értékét a beírt karakterek ASCII-kódjai határozzák meg. Mint láttuk, az aposztrófok között megengedett az idézőjelek használata, és ez fordítva is igaz. Ha azonban a stringben a határoló jellel megegyező karaktert akarunk szerepeltetni, akkor az adott jelet meg kell duplázni:

'John"s cat'

9.2.1 D (dump) - memóriatartalom kiíratása

A D (dump, **tárkiíratás) parancs segítségével tekinthetjük meg a képernyőn a memória kijelölt részét.** A parancs általános formája:

D [*cím*]

vagy

D [*intervallum*]

Ha a parancsot paraméter nélkül adjuk ki, akkor a DEBUG az előző D parancs által kiírt utolsó byte utáni címet tekinti kezdőcímnek. Ha csak a relatív cím (offset) részét írtuk ki, akkor a DS regiszter tartalma adja a szegmenscímet. Ha nem volt megelőző D parancs, akkor DS:100-tól kezdődik a hívás.

Ha a parancsban egy intervallumot specifikáltunk, akkor az intervallumhoz tartozó byte-okat jeleníti meg a DEBUG. Ha csak a kezdőcímet adtuk meg (explicit vagy implicit módon), akkor hexadecimálisan 80 (decimálisan 128) byte íródik ki a képernyőre.

A képernyő baloldalán (a cím mellett) az egyes byte-ok hexadecimális formában, a jobboldalon pedig ASCII-karakterek formájában jelennek meg. A nyomtathatatlan karakterek helyett egy-egy pont látszik. Minden sorban tizenhat byte szerepel (kivéve az esetleges első és utolsó csonka sort). A baloldalon, a nyolcadik és a kilencedik byte közé egy kötőjelet tesz a DEBUG.

A példánkban a DS regiszter által meghatározott szegmens hexadecimális 100-tól 17F-ig terjedő relatív címeinek adattartalmát íratjuk ki. Az alábbi három parancs ugyanezt a feladatot hajtja végre:

```
D
D DS:100 17F
D 100 L 80
```

Ha a tárképet nyomtatásban is látni akarjuk, használjuk a <Shift-PrtSc>, a <Ctrl-P> vagy a <Ctrl-PrtSc> billentyűkombinációkat.

9.2.2 R (register) - regiszterek kiíratása, megváltoztatása

Az R (regiszter) parancs lehetővé teszi a mikroprocesszor regisztereinek és flag-jeinek (jelzőbitjeinek) a megtekintését és módosítását. A parancs általános formája:

R [*regiszternév*]

ahol a *regiszternév* az alábbi nevek egyike lehet:

AX, BX, CX, DX, SP, BP, SI, DI, DS, ES, SS, CS, IP, PC, F.

Az **IP** (Instruction Pointer) és a **PC** (Program Counter) rövidítések ugyanarra a regiszterre utalnak. Az **F** rövidítés a flag-eket (jelzőbiteket) jelenti.

Ha az R parancsot paraméter nélkül adjuk ki, akkor a DEBUG megjeleníti a képernyőn a regiszterek és a flag-ek tartalmát, valamint a következő végrehajtandó utasítás kódját és mnemonikus alakját. A regiszterek (AX, BX, CX, DX, SP, BP, SI, DI, DS, ES, SS, CS, IP) tartalma hexadecimálisan íródik ki. Az egyes flag-ek értéke kétbetűs rövidítések formájában jelenik meg. A rövidítések magyarázatát az alábbi táblázat tartalmazza:

Flag neve	**Beállítva (1)**	**Törölve (0)**
Túlcsordulás (overflow) igen/nem	OV	NV
Irány (direction) csökkenő/növekvő	DN	UP
Megszakítás (interrupt) engedélyezve/letiltva	EI	DI
Előjel (sign) negatív/pozitív	NG	PL
Nulla (zero) igen/nem	ZR	NZ
Belső átvitel (auxiliary carry) igen/nem	AC	NA
Paritás (parity) páros/páratlan (even/odd)	PE	PO
Átvitel (carry) igen/nem (yes/no)	CY	NC

A következő végrehajtandó utasításról a DEBUG az alábbi adatokat közli:

- az utasítás kezdőcímét (ezt a CS:IP regiszterpár tartalmazza),
- hexadecimális kódját és
- visszafordított (disassemblált) alakját (azaz mnemonikját és operandusait).

Ha az R parancsban valamelyik regiszter nevét is megadtuk, akkor a DEBUG csak a megnevezett regiszter tartalmát írja ki (hexadecimálisan). Ezután - a következő sorban - egy kettőspont jelenik meg, amelyik mellé a regiszter új értékét írhatjuk be (egy maximum négyjegyű hexadecimális szám formájában). Ha ehelyett csak az <ENTER>-t nyomjuk le, akkor a regiszter tartalma nem változik.

Ha az R parancsban az F paramétert használjuk, akkor a DEBUG csak a flag-ek értékét jeleníti meg, a fenti táblázat rövidítéseivel. A kódok mellett egy kötőjelet látunk, emellé írhatjuk a megváltoztatni kívánt flag-ek értékeit (tetszőleges sorrendben, szintén a fenti rövidítéseket használva). A flag-ek kódjai között nem kötelező szóközt hagyni. Ha változtatás helyett csak az <ENTER>-t nyomjuk le, a jelzőbitek tartalma nem módosul.

Nézzünk meg néhány példát az R parancs használatára:

```
R
```

A képernyőn megjelenik a regiszterek tartalma és a flag-ek értéke, valamint a soron következő utasítás.

```
R DX
```

A képernyőn a DX regiszter tartalmát látjuk, például ezt:

```
DX 14D2
:1C2
```

Az alábbi példában a flag-eket vizsgáljuk meg:

```
R F
```

A képernyőn ezt látjuk:

```
OV UP EI PL NZ AC PE NC-CY NV
```

9.2.3 F (fill) - memória feltöltése

Az F (fill, feltöltés) parancs segítségével tetszőleges adattal tölthetjük fel a RAM memória kijelölt részét. A parancs általános alakja:

F *intervallum adatlista*

Az *intervallum* és az *adatlista* fogalmát a 9.2 pont bevezető részében ismertettük. Ha az *intervallum* kezdőcíme nem jelöl ki szegmenst, akkor a DS regiszter tartalma határozza meg a szegmens címét. A parancs hatására a DEBUG feltölti a kijelölt tartományt az *adatlistában* felsorolt adatokkal.

Ha az *adatlista* több byte-ot tartalmaz, mint az intervallum, akkor a felesleges byte-ok nem kerülnek be a memóriába. Ha az intervallum hosszabb az *adatlistánál*, akkor az *adatlista*

elemei mindaddig ismétlődnek a tárban, ameddig az intervallum végéhez nem érünk.

Az alábbi példában a DS által meghatározott szegmens hexadecimális 200-tól 2FF-ig terjedő részét töltjük föl nullákkal:

F 200 2FF 0

A következő parancs hatására a kijelölt intervallum (10000:0-tól 5F-ig) egy hat-byte-os listával töltődik fel. A lista elemei tizenhatszor ismétlődnek a tárban:

F 10000:0 L 60 2 F1 'EZ' 10 0

A memóriába a string karaktereinek ASCII-kódja kerül be hexadecimális formában, a lista tehát a következő byte-okból áll: 02, F1, 45, 5A, 10, 00.

9.2.4 E (enter) - memóriatartalom megváltoztatása

Az E (enter, adatbevitel) parancs kiadásával lehet a RAM memória tetszőleges byte-jait megváltoztatni. A parancs általános alakja:

E *cím* [*adatlista*]

A *cím* és az *adatlista* specifikációját a 9.2 pont bevezető részében ismertettük. Ha a *cím* csak egy relatív címből (offset-ből) áll, akkor a DS regiszter tartalma határozza meg a szegmens kezdőcímét.

Ha a parancsban nem adtunk meg *adatlistát*, akkor interaktív (párbeszédes) üzemmódban módosíthatjuk a tár tartalmát. A parancs hatására a DEBUG kiírja a sor elejére a megadott címet, és mellé az ott található byte hexadecimális értékét. A szám után egy pont jelenik meg. Ezután rajtunk van a sor, és a következő lehetőségek közül választhatunk:

- A szóköz-billentyű lenyomásával változatlanul hagyjuk a kiírt byte-ot. Ekkor a DEBUG a következő byte értékét mutatja.
- Új értéket adunk a megjelenített byte-nak, egy maximum kétjegyű hexadecimális szám beírásával. Ezután a szóközbillentyűvel léphetünk a következő, vagy a kötőjellel az előző byte-ra.
- A kötőjel lenyomására a DEBUG az előző cím tartalmát mutatja meg (új sorban). Ezt is módosíthatjuk.
- Ha lenyomjuk az < ENTER > -t, kilépünk az E parancsból.

Ha az E parancsban egy *adatlistát* is megadtunk, akkor a lista egyes byte-jainak tartalma átmásolódik a memóriába, az előírt címtől kezdve.

Az első példában az interaktív üzemmódot mutatjuk be:

E ES:2A0

A képernyőn a következőket látjuk:

0621:02A0 1F.

A megjelenített byte tartalmát 5B-re módosítjuk, majd a szóköz-billentyű lenyomásával tovább lépünk:

0621/02A0 1F.5B 41.

A 2A1-en elhelyezkedő byte-ot nem változtatjuk meg, hanem a szóköz-billentyűvel a következő byte-ra térünk át:

0621:02A0 1F.5B 41. 4A.

A 2A2-es byte-ot 4A-ról 00-ra írjuk át, majd az <ENTER>-rel kilépünk az E parancsból:

0621:02A0 1F.5B 41. 4A.00

Az alábbi példában a DS regiszter által kijelölt szegmens 100-tól 106-ig terjedő területére írjuk be az adatlista elemeit:

E 100 "PC DOS",0

9.2.5 C (compare) - memóriaterületek összehasonlítása

A C (compare, összehasonlítás) parancs byte-onként összehasonlítja a tár kijelölt területeit és jelzi a különbségeket. A parancs általános alakja:

C *intervallum cím*

Az *intervallum* és a *cím* fogalmát a 9.2 pont bevezető részében ismertettük. Ha a cím (vagy az intervallum kezdőcíme) csak egy relatív címből (offset-ből) áll, akkor a DEBUG a DS regiszter tartalmát tekinti a hiányosan specifikált szegmens kezdőcímének.

Az első memóriaterületet az intervallum jelöli ki, a másodiknak csak a kezdőcímére van szükség, mert az összehasonlítandó byte-ok számát az első terület hossza határozza meg.

Az eltéréseket a DEBUG az alábbi formában jelzi a képernyőn:

cím1 byte1 byte2 cím2

ahol cím1 és cím2 a két összehasonlított byte címe, byte1 és byte2 pedig azok tartalma.

A példában a CS:100-tól 14F-ig terjedő byte-okat hasonlítjuk össze a DS:100-tól kezdődő byte-okkal. Az intervallum megadásánál az L paramétert használjuk (a hossz kijelölésére):

C CS:100 L 50 DS:100

A DEBUG - a COMP paranccsal ellentétben - az összes eltérő byte-ot megmutatja. Ha a COMP parancs nem megfelelő két file összehasonlítására (pl. eltérőek a hosszak, ill. 10-nél több eltérés van és ezeket is látni akarjuk) a következőket tehetjük:

- behívjuk a DEBUG programot,
- betöltjük a memóriába az első file-t,
- betöltjük a második file-t,

• összehasonlítjuk őket a C (compare) paranccsal.

Példa: tegyük fel, hogy mindkét file mérete 64 Kbyte-nál kisebb, akkor az összehasonlítás a következőképpen mehet:

```
DEBUG
- F 2000:0 L 0 0
- F 3000:0 L 0 0
- N [d:][út]file1[.kit]
- L 2000:0
- N [d:][út]file2[.kit]
- L 3000:0
- C 2000:0 L 0 3000:0
```

Ha túl sok az eltérés, a listát a képernyőn megállíthatjuk <Ctrl-S>-sel, ill. nyomtatóra is kiírathatjuk, ha a C parancs előtt <Ctrl-Prtsc>-t nyomunk le.

Megjegyzés: amennyiben .COM vagy .EXE file-okat akarunk így összehasonlítani, előtte át kell nevezni azokat más kiterjesztésre!

9.2.6 M (move) - memóriaterület átmásolása

Az M (move, mozgatás) parancs átmásolja a memória kijelölt területét a megadott címre. (Az eredeti tárterület eközben nem változik, kivéve, ha a másolás cél-területe átfedi az eredeti intervallumot.) A parancs általános alakja:

M *intervallum cím*

Az *intervallum* és a *cím* megadási módjait a 9.2 pont bevezető részében ismertettük. Ha a cím (vagy az intervallum kezdőcíme) csak egy relatív címből (offset-ből) áll, akkor a DEBUG a DS regiszter tartalmát tekinti a meg nem határozott szegmens kezdőcímének.

9.2.7 S - karaktersorozat keresése

Az S (search, keresés) parancs adott memóriatartományban egy adott karaktersorozat összes előfordulási helyét megkeresi. A parancs alakja a következő:

S *intervallum adatlista*

Az *intervallum* és az *adatlista* fogalmát a 9.2 pont bevezető részében ismertettük. Ha az *intervallumban* nem adjuk meg a szegmens értékét, a program a DS regiszterben levő értéket használja. Például:

```
S CS:100 1500 "HELLO"0
```

parancs az aktuális programszegmens 100-1500 címei közti részt nézi végig a "HELLO"0 karaktersorozat után kutatva. Minden egyes előfordulás után kiírja a "H" betű helyét. Ha egyszer sem fordul elő az adott intervallumban, akkor semmilyen üzenetet sem kapunk.

9.2.8 Q - kilépés a programból

A Q (quit) parancs kiadása után a DEBUG befejezi működését és visszaadja a vezérlést a DOS-nak (vagy annak a programnak, amelyik a DEBUG-ot elindította). Ha a vizsgált programra szükségünk van, azt a W paranccsal a Q kiadása előtt el kell menteni.

9.2.9 A (assemble) - assemblálás

Az A (assemble, assemblálás) parancs segítségével assembly nyelvű utasításokat írhatunk. A DEBUG előállítja az utasítások kódját, és elhelyezi a memóriában. A parancs általános alakja:

A [*cím*]

A cím specifikálási módját a 9.2 pont bevezető részében ismertettük. Ha a címet nem írjuk ki és az első A parancsnál tartunk, akkor az assemblálás a CS:IP0 címen kezdődik. A későbbiekben, ha nem adunk meg címet, akkor az előző A parancs által tárolt utasítás utáni byte-nál folytatódik a kódgenerálás. Ezt az alapértelmezést bármikor módosíthatjuk, ha az A parancsban a konkrét címet is megadjuk.

A parancs kiadása után a DEBUG megjeleníti a képernyőn az aktuális kezdőcímet. Akkor beírhatunk egy assembly nyelvű utasítást, amely a 8088-as (vagy 8086-os) processzor, illetve a 8087-es segédprocesszor utasításkészletébe tartozik. Ha szintaktikai hibát vétettünk, akkor az "Error" (hiba) üzenetet kapjuk, és a DEBUG újra kiírja az aktuális címet.

Az egyes assembly utasítások beírása után nyomjuk le az <ENTER> billentyűt. Ekkor az utasítás kódja bekerül a tárba, és a DEBUG a következő memóriacímet mutatja. Ha nem kívánunk több utasítást kiadni, a cím után közvetlenül nyomjuk le az <ENTER>-t, így térhetünk vissza a DEBUG parancs-szintjére.

Az assembly utasítások beírásánál a következő szabályokat kell betartani:

- A numerikus értékeket csak hexadecimálisan írhatjuk, maximum négy számjeggyel.
- Az ismétlő (REP...), szegmenskijelölő (CS:, DS:, ES:, SS: stb.) prefixeket a hozzájuk tartozó utasítás elé kell írni (azzal egy sorba, vagy az előző sorba).
- A string-kezelő utasításokban közölni kell a string méretét. A MOVS utasítást például MOVSB, illetve MOVSW formában kell írnunk, attól függően, hogy egy byte-ot vagy egy szót (word) akarunk mozgatni.
- A távoli visszatérés (far return) mnemonikja: RETF.
- A DEBUG az ugró utasításokban megadott címek elhelyezkedése alapján dönti el, hogy rövid (2 byte-os), közeli (3 byte-os) vagy távoli (5 byte-os) ugró utasítást generáljon-e. Ezt az automatizmust kikapcsolhatjuk, ha a NEAR (közeli), illetve a FAR (távoli) prefixet írjuk a cím elé. A NEAR szót NE-re lehet rövidíteni. Példák:

```
0A27:0100 JMP 12C          ; rövid ugrás (2 byte-os)
0A27:0100 JMP NEAR 12C     ; közeli ugrás (3 byte-os)
0A27:0100 JMP FAR 12C      ; távoli ugrás (5 byte-os)
```

- A DEBUG nem tudja megkülönböztetni, hogy az egyes operandusok byte- vagy szó- (word) hosszúságú memóriaterületekre hivatkoznak-e. Ezért - ahol mindkét típusú hi-

vatkozás megengedett - használjuk az operandusok előtt a BYTE PTR (rövidítve: BY), illetve a WORD PTR (rövidítve: WO) prefixeket.

Példák:

```
INC WORD PTR [200]
NOT BY [DI]
```

- A DEBUG mini-assemblere a közvetlen (immediate) operandusoknál nem vár el semmilyen megkülönböztető jelet. Így a

  MOV BH,1A

  utasítás hatására a BH regiszterbe a hexadecimális 1A közvetlen érték kerül. A tárcímekre való hivatkozást viszont szögletes zárójelek közé kell tennünk:

  MOV BH,[1A]

  Ennél az utasításnál a BH-ba a 001A memóriacím tartalma töltődik be.

- A DEBUG megengedi a DB (define byte, byte definiálása) és a DW (define word, szó definiálása) parancsok használatát, amelyek helyet foglalnak a tárban, és ott helyezik el a megadott adatlista elemeit. Példák:

```
DB 2B,0,'Szöveg',FF
DW 12A4,8A6
```

- Az indirekt címzési módszereket a megszokott módon alkalmazhatjuk:

```
ADC AX,2A[BP+3].[SI-1]
POP [SI]
PUSH [BP+DI]
```

- Egyes utasításokra az assemblerek két szinonim mnemonikot is alkalmaznak (például a JL utasítás megegyezik a JNGE-vel, a LOOPNE a LOOPNZ-vel). A DEBUG e szinonimákat is elfogadja.

- A 8087-es aritmetikai segédprocesszor utasításai előtt adjuk meg a WAIT vagy a FWAIT prefixet. Például:
  FWAIT FADD ST,ST(3)

Nézzünk meg végül egy rövid példát az A parancs használatára. Az assemblálás kezdőcíme legyen hexadecimális 1000. Írjuk be:

A 1000

A baloldalon látható címeket a DEBUG írja ki, az utasítások begépelése a mi feladatunk. Az utolsó sorban (a 07F1:1004-es cím mellett) csak az <ENTER>-t nyomtuk le, így léptünk ki az A parancsból:

```
07F1:1000 POP DX
07F1:1001 MOV AX,DX
```

```
07F1:1003 RET
07F1:1004
```

9.2.10 U (unassemble) - disassemblálás

Az U (unassemble, disassemblálás) parancs a memóriában található gépi kódú utasításokat assembly utasítások formájában jeleníti meg a képernyőn. A parancs általános alakja:

U [*cím*]

vagy

U [*intervallum*]

A *cím* és az *intervallum* megadásának módjait a 9.2 pont bevezető részében ismertettük. Ha a parancsot paraméter nélkül adjuk ki, akkor a legutóbbi U parancs által kiírt utasítás utáni byte-tól kezdődik a disassemblálás (így lehet egy hosszabb tárterületet a legegyszerűbben áttekinteni). Ha a DEBUG indítása óta most adtunk ki először U parancsot, akkor a kezdőcím a CS:IP0 lesz. Egyébként is, ha az *intervallum*ban vagy a *cím*ben nem nevezünk meg szegmensregisztert (vagy szegmenscímet), akkor a CS regiszter által mutatott szegmens tartalmát dolgozza föl az U parancs.

Ha a parancs második formáját alkalmazzuk (*intervallum* megadásával), akkor a DEBUG a kijelölt tárterületet disassemblálja. Ha csak egy *cím*et írtunk be, vagy az alapértelmezés szerinti címet használjuk, akkor a kezdőcímtől indulva harminckét byte disassemblálása történik meg. Az utasítások változó hossza miatt a program esetenként néhány byte-tal többet is feldolgoz az előírtnál (ha az utolsó utasítás túlnyúlik az intervallum határán).

A képernyőn a következő adatok jelennek meg: az utasítások címe, hexadecimális kódja és assembly nyelvű alakja. Az

U 100 L 6

parancsra például az alábbi választ kaphatjuk:

```
07D2:0100 1E       PUSH DS
07D2:0101 2BC0      SUB AX,AX
07D2:0103 B402      MOV AH,2
07D2:0105 206472    AND [SI + 72],AH
```

Ugyanezt az eredményt kaptuk volna az L 7 vagy az L 8 hosszak megadásával, vagy az

U 100 107

parancs kiadásakor is.

Ha a képernyőre nem férnek ki a DEBUG által kiírt adatok, a listázást a <Ctrl-NumLock> billentyűk lenyomásával félbeszakíthatjuk, majd bármely más billentyűvel újraindíthatjuk.

9.2.11 G (go) - program futtatása

A G (go, futtatás) parancs a memóriában található gépi kódú programot indítja el. A parancs általános alakja:

G [=*cím*][*cím* [*cím*]...]

A G parancs kiadásakor a vezérlés az egyenlőségjel után írt *cím*re adódik át (az alapértelmezés szerinti szegmens a CS regiszter által meghatározott kódszegmens). Ha ezt a címet nem adtuk meg, akkor (szintén a kódszegmensben) az utasításmutató (IP) regiszterben található címnél kezdődik a futtatás.

A G parancsban egyenlőségjelek nélkül írt *cím*ek (legfeljebb tizet sorolhatunk föl) a megszakítási pontokat (breakpoint-okat) jelölik ki. Ha a megszakítási pontoknál nem adtunk meg szegmenst, akkor a CS szegmens lesz az alapértelmezés. Amikor a program ezekhez a pontokhoz ér, a futás félbeszakad, és megvizsgálhatjuk a regiszterek vagy a memória tartalmát. Szétágazó program esetében több megszakítási pontot érdemes kijelölni az egyes ágakon, így megálláskor látni lehet, hogy a program melyik ága futott. Ez a lehetőség a programok tesztelésénél fontos szerepet játszik.

A megszakítási pontokon a DEBUG a program indítása előtt kicseréli az eredeti kódot egy speciális megszakító utasítás kódjával. Amikor a végrehajtás során egy megszakítási ponthoz érünk, a DEBUG visszaállítja valamennyi megszakítási ponton az eredeti utasításkódokat, majd a képernyőn - az R parancsnál leírt formában - megjeleníti a regiszterek és a flag-ek tartalmát, valamint a soron következő utasítást. Vigyázat: ha a program egyik megszakítási ponthoz sem ér el, akkor az eredeti utasítások visszaállítása nem történik meg!

A megszakítási pont elérése után egyszerűen lehet továbbfuttatni a programot: a G parancsban ne adjunk meg egyenlőségjeles címet(megszakítási pontokat kijelölhetünk). Az alapértelmezés szerinti CS:IP kezdőcím pontosan a megszakítási utasításra mutat.

Ha nem definiáltunk megszakítási pontokat (vagy a végrehajtás során ezekhez nem jutottunk el) és a program a futása végén visszaadja a vezérlést a DEBUG-nak, akkor a "Program terminated normally" (a program futása normálisan fejeződött be) üzenetet kapjuk. Ha a programot ezután ismét futtatni akarjuk, általában újra be kell töltenünk a lemezről (lásd az N és az L parancsokat).

A G parancs kiadásakor a veremben legalább hat byte szabad területnek kell lennie, különben a futtatás nem lehetséges.

Az első példában a már korábban betöltött programunkat futtatjuk le. A program kezdőcíme: 2F3. Megszakítási pontokat nem jelölünk ki:

```
G =2F3
```

A második példában a program az (alapértelmezés szerinti) 100-as címen kezdődik. Két megszakítási pontot definiálunk, mert kíváncsiak vagyunk a regiszterek és a tár tartalmára, amikor a program az 54A, illetve az 101C címekhez ér:

```
G 101C 54A
```

A megszakítási pontok beírásának sorrendje közömbös. Tegyük föl, hogy a program elju-

tott az 54A címhez. Ott a végrehajtás félbeszakad, a DEBUG visszaállítja mindkét megszakítási ponton az eredeti utasításkódokat, majd megjeleníti a regiszterek és a flag-ek tartalmát, valamint a következő utasítást.

9.2.12 T (trace) - lépésenkénti végrehajtás

A T (trace, nyomkövetés) parancs a memóriában található gépi kódú programot futtatja. A végrehajtás minden, a parancsban megadott számú utasítás végrehajtása után megszakad. A parancs általános alakja:

T [=*cím*] [*szám*]

ahol a futtatás kezdő*címét* az egyenlőségjel után lehet megadni (a 9.2 pont bevezető részében leírtak szerint), a *szám* helyére pedig egy maximum négyjegyű hexadecimális számot írhatunk (ez jelöli ki az egy menetben végrehajtandó utasítások számát).

Ha a kezdőcímet nem adjuk meg, a DEBUG a CS:IP regiszterpárban található címen kezdi el a futtatást. Ha csak egy relatív címet (offset-et) írunk be, akkor az a CS regiszter által meghatározott szegmensen belül érvényesül.

A szám megadása nem kötelező a T parancsban. Ha kihagyjuk, az alapértelmezése: 1.

A T parancs minden egyes utasítás végrehajtása után megjeleníti a képernyőn a regiszterek és a flag-ek tartalmát, valamint a következő utasítást (az R parancsnál ismertetett formában). Ha a kiírás túl gyors, bármikor félbeszakíthatjuk a <Ctrl-NumLock> billentyűkkel. A listázás bármelyik billentyű lenyomására folytatódik.

A T parancsban megadott számú utasítás végrehajtása után a DEBUG megáll és új parancsra vár. Ha ezután egy cím nélküli T parancsot adunk ki, akkor a program futtatása pontosan a félbeszakítás helyén folytatódik.

A példában a kódszegmens 20B címétől indítjuk a programot, és tíz (hexadecimálisan: A) utasítás végrehajtását kérjük:

```
T =20B A
```

A DEBUG mind a tíz utasítás végrehajtása után megjeleníti a regiszterek és a flag-ek tartalmát, valamint a soron következő utasítást. A program továbbfuttatásához az alábbi parancsot adjuk ki:

```
T A
```

Ezután - ha a programot tíz utasításonként tovább akarjuk futtatni - már csak az <F3> és az <ENTER> billentyűvel kell ismételni az előző parancsot.

9.2.13 P - utasítássorozat végrehajtása

A P (proceed) parancs hasonlóan működik, mint a T (trace) parancs, azzal a lényeges különbséggel, hogy bizonyos gépi kódú utasításokat egyetlen egynek tekint. Ezek a következők:

- szubrutinhívás;
- ciklusutasítás;
- megszakítás;
- ismételt string utasítás.

Az utasítás általános alakja:

P [=*cím*][*szám*]

Ha a memóriában a következő program található:

```
0100 CALL 1000
0103 INT 214
0106 JZ 3000
```

akkor a P2 utasítás hatására a programnak a CALL 1000 és CALL 2000 parancsai hajtódnak végre, függetlenül attól, hogy az 1000-nél kezdődő alprogram és a 21-es megszakítást kezelő alprogram valójában hány utasításból áll.

A P paranccsal lehetőségünk van a már letesztelt és ellenőrzött programrészek nyomkövetésének kikapcsolására. A P parancs csak a 3.xx verziókban használható.

9.2.14 N (name) - a file kijelölése

Az N (name, név) parancs segítségével hozhatunk létre file-vezérlő blokkokat (File Control Block), és specifikálhatjuk a file-neveket az L (load) és a W (write) parancsok előtt.

A parancs általános alakja:

N *filespec* [*filespec*] ...

ahol a *filespec* a szokásos [*d:*][*út*]*file-név*[*kit*] formában adható meg.

A DEBUG az első két file részére a DS:5C, illetve a DS:6C címeken hoz létre file-vezérlő blokkot. Az N parancs után írt valamennyi karakter (a file-specifikációkat is beleértve) átmásolódik a DS:81 címtől kezdve, a paraméter-területre.

A DS:80 címre a beírt karakterek száma kerül.

Ha a DEBUG indító parancsában már megneveztünk egy file-t, akkor a DS:5C címen elhelyezkedő file-vezérlő blokk az ott kijelölt file adatait tartalmazza. Az ezután kiadott L és W parancsok hatására is ugyanennek a file-nak a beolvasása, illetve kiírása fog megtörténni.

Ha a DEBUG indításakor nem adtunk meg file-nevet, vagy pedig nem az ott kijelölt file-t akarjuk olvasni (írni), akkor az L (ill. W) parancs előtt az N paranccsal kell az új file-t specifikálni.

A már betöltött programok esetében az N parancs segítségével tölthetjük föl a program file-vezérlő blokkjait és paraméterterületét a szükséges információkkal.

Tegyük fel, hogy a DEBUG indító parancsában nem adtunk meg file-nevet. Most viszont be akarjuk tölteni a PROBA nevű file-t a tárba. Az alábbi parancsokat kell kiadnunk:

```
N PROBA
L
```

A második példában feltételezzük, hogy a DEBUG indításakor már megtörtént a PROBA program betöltése. Most a program két file-vezérlő blokkját és paraméter-területét töltjük fel a kívánt adatokkal:

```
N ELSO.DAT MASODIK.TXT PARAM1 PARAM2
```

A DEBUG kialakítja a DS:5C és DS:6C című file-vezérlő blokkokat az ELSO.DAT és a MASODIK.TXT file-ok adataival (DS:5C-n eddig a PROBA file vezérlő blokkja helyezkedett el). A DS:80-on a paraméterekként megadott karakterek darabszáma (35, azaz hexadecimális 23) áll, DS:81-től kezdve pedig maguk a paraméterek (az N utáni szóköztől egészen a PARAM2 végéig).

9.2.15 L (load) - file (kijelölt szektorok) betöltése

Az L (load, betöltés) parancs betölti a memóriába a kijelölt file-t, illetve a megadott lemezes szektorok tartalmát. A parancs általános alakja:

L [*cím* [*meghajtó szektor szám*]]

ahol

cím megadása a 9.2 pont bevezető részében leírtak szerint történhet,

meghajtó helyébe egy egyjegyű számot lehet írni (a 0 szám az "A", az 1 a "B" lemezegységre utal stb.),

szektor a beolvasni kívánt első szektor sorszáma (0-tól számozva). A 0. sorszám a 0. sávon a 0. fej által beolvasott 1. szektort jelenti. A sorszámozás a 0. sávon és a 0. fej pályáján folytatódik a 2., 3. stb. szektorokkal. Ezután - ugyanezen a sávon - az 1. fej alatt elhelyezkedő szektorok kerülnek sorra. Amikor a 0. sávon valamennyi fejet számba vettük, áttérünk az 1. sávra és így tovább.

szám a beolvasandó szektorok száma, hexadecimálisan (maximális értéke: hexadecimális 80).

Ha az L parancsot paraméterek nélkül adjuk ki, vagy csak a címet írjuk be, akkor a DEBUG a DS:5C címen elhelyezkedő file-vezérlő blokkban (FCB-ben) specifikált file-t tölti be a tárba. Az FCB-t az N paranccsal lehet feltölteni, de ha ezt nem használtuk, akkor ott a DEBUG indító parancsában megnevezett file adatai szerepelnek.

A file-ok betöltése a DS:100 címtől történik, ha a parancsban ettől eltérő címet nem adtunk meg. A .COM és az .EXE kiterjesztésű file-okat a DEBUG - az általunk beírt címtől függetlenül is - a CS:100-ra tölti be. Ha az EXE file szerkesztésekor (LINK) a program magas címre való betöltését írtuk elő, akkor ezt az utasítást a DEBUG is betartja.

A file beolvasása után a BX és a CX regiszterek a file hosszát mutatják. Az .EXE és a .HEX kiterjesztésű file-ok betöltéséről, valamint a BX és a CX regiszterek tartalmáról a 9.1 pontban találunk bővebb leírást.

Ha az L parancsban lemezes szektorok beolvasását kértük, akkor a betöltés kezdőcímét is meg kell adnunk. Amennyiben ez csak egy relatív címből (offset-ből) áll, a DEBUG a CS regiszter által mutatott szegmenst használja. A kijelölt szektorok tartalma ettől a címtől kezdve, folyamatosan tölti ki a memóriát.

Az alábbi példában a PACMAN.COM programot töltjük be a CS:100 címre. Ha a DEBUG indító parancsában a program neve már szerepelt (DEBUG PACMAN.COM), és N parancsokat nem használtunk, akkor elég ennyit írni:

```
L
```

vagy

```
L 100
```

Ha az indító parancsban a PACMAN.COM nevét nem adtuk meg, akkor az N parancsra is szükség van:

```
N PACMAN.COM
L
```

Az utolsó példában a "B" egységben levő kétoldalas floppy lemezről olvassuk be a tárba az 1. sáv, 0. oldal (azaz: 0. fej) valamennyi szektorának adatait. A sávon kilenc szektor található, az első szektor sorszáma: hexadecimális 12. A betöltés kezdőcíme legyen 1000:0:

```
L 1000:0 1 12 9
```

A szektorok egyenként 512 (hexa 200) byte-ot tartalmaznak, így a memóriában az 1000:0-tól az 1000:11FF-ig terjedő tartományt töltöttük fel adatokkal.

Megjegyzés:

Fontos tudni, hogy a floppy lemezekkel ellentétben a winchester-eknél a szektorszámozás a DOS partíció kezdetétől relatív (nem a winchester fizikai első cilinderének első sávjától), így DEBUG alól - az L paranccsal - nem tudjuk pl. a winchester partíció-táblát (0. cilinder 0. oldal 1. szektorát) beolvasni.

9.2.16 W (write) - file (kijelölt szektorok) kiírása

A W (write, kiírás) parancs kiírja a memória tartalmát egy file-ba, illetve a megadott lemezes szektorokba. A parancs általános alakja:

W [*cím* [*meghajtó szektor szám*]]

ahol a paraméterek jelentése megegyezik az L parancsnál leírtakkal (azzal az értelemszerű különbséggel, hogy itt nem beolvasásról, hanem kiírásról van szó).

Ha a W parancsot paraméterek nélkül adjuk ki, vagy csak a címet írjuk be, akkor a DEBUG a DS:5C címen elhelyezkedő file-vezérlő blokkban (FCB-ben) specifikált file-ba írja ki a tár tartalmát. Az FCB-t az N paranccsal lehet feltölteni, de ha ezt nem használtuk, akkor ott a DEBUG indító parancsában megnevezett file adatai szerepelnek.

A file-ba való kiírás a memória CS:100 címén kezdődik (kivéve, ha a parancsban ettől eltérő címet adtunk meg). A kiírt byte-ok számát a BX és a CX regiszterekben szereplő érték határozza meg (lásd a 9.1 pontban). Ha a DEBUG indítóparancsában előírtuk egy fi-

le betöltését, akkor a BX és a CX regiszterek kezdetben felveszik a file hosszának megfelelő értéket. Ha e regiszterek tartalmát nem módosítottuk, és a file hossza sem változott, akkor minden további nélkül kiadhatjuk a W parancsot. Egyéb esetben először korrigálnunk kell a BX és a CX regiszterek értékét.

Egy megszorítás: az .EXE és a .HEX kiterjesztésű file-okat nem írhatjuk vissza a W paranccsal, mivel a DEBUG ezeket a file-szerkezeteket nem tudja előállítani. Ha mégis ilyen kiterjesztésű file-t akarunk módosítani, akkor - a betöltés előtt - nevezzük át az adott file-t más kiterjesztésűre. A file-t ezután módosíthatjuk a DEBUG-gal, de futtatni (a G vagy a T paranccsal) nem tudjuk. A lemezre való kiíratás után visszaadhatjuk a file-nak az eredeti nevét.

Ha a W parancsban szektorokba való kiíratást kértünk, akkor a kiírandó tárterület kezdőcímét is meg kell adnunk. Amennyiben ez csak egy relatív címből (offset-ből) áll, a DEBUG a CS regiszter által mutatott szegmenst használja. A memória tartalma folyamatosan íródik ki ettől a címtől kezdve, amíg a megadott számú szektor fel nem töltődik.

Az alábbi példában az XYZ.DAT file-t írjuk vissza a lemezre. Ha a file-t a DEBUG indító parancsával, vagy a legutóbbi L paranccsal töltöttük be a CS:100 címre (továbbá a BX és a CX regiszterek tartalma, valamint a file hossza azóta nem változott meg), akkor elég az alábbi parancsot kiadni:

```
W
```

Ha a fenti feltételek nem teljesülnek, akkor először beírjuk az R paranccsal a BX és a CX regiszterekbe a file hosszát. Legyen a hossz hexadecimális 1000. Ekkor a BX-be nullát, a CX-be pedig az 1000 értéket írjuk. Ezután adjuk ki az alábbi parancsokat (a file-t a "C" meghajtóra írjuk):

```
N C:XYZ.DAT
W
```

A következő példában a kódszegmens 100-tól 4FF-ig terjedő részét írjuk ki konkrétan megnevezett szektorokba. (Vigyáznunk kell, hogy ezzel ne tegyük tönkre a lemezen tárolt file-okat!) A fenti memóriaterület hossza decimálisan 1024 byte, tehát két szektorban fér el. A kiírás kezdődjön az "A" egységben levő (kétoldalas) lemez második sávján, a nulladik oldalon, a harmadik szektorban:

```
W 100 0 26 2
```

A 100 tehát a kiírandó terület relatív kezdőcíme, a 0 a lemezegység száma (az "A" jelű floppy meghajtóé), a 26 az első írandó szektor sorszáma, a 2 pedig a kiírandó szektorok száma. A szektor sorszámát az alábbi táblázat magyarázza:

Sáv	Fej	Szektor	Szektor sorszáma
0	0	1	0
0	1	1	9
1	0	1	12
1	1	1	1B
2	0	1	24
2	0	3	26

9.2.17 H - hexadecimális műveletvégzés

A H (hexadecimális aritmetika) parancs segít a DEBUG parancsaihoz szükséges hexadecimális értékek kiszámításában. A parancs általános alakja:

H *szám szám*

ahol a két *szám* helyére legfeljebb négyjegyű, hexadecimális értékeket kell írni.

A H parancs hatására a DEBUG megjeleníti a képernyőn a két szám összegét és különbségét (szintén hexadecimális formában).

A példában a 2A31 címen kezdődő, 1BF hosszúságú string utolsó karakterének címét szeretnénk megtudni. A két szám összege már a string utáni byte-ra mutatna, ezért a string hosszából egyet levonunk:

H 2A31 1BE

A DEBUG válasza:

2BEF 2873

Az első érték adja a keresett címet, a második érték (a két szám különbsége) jelen esetben nem fontos számunkra.

9.2.18 I - input a kijelölt port-ról

Az I (input, adatbevitel) parancs beolvas egyetlen byte-ot a megadott című port-ról (a processzorhoz kapcsolt adatcsatornáról), és megjeleníti a képernyőn a beérkezett byte hexadecimális értékét. A parancs általános alakja:

I *portcím*

ahol a *portcím* egy maximum négyjegyű hexadecimális szám lehet (0-tól FFFF-ig).

A példában a 320-as című port-ról olvasunk, az eredményt (a 3A értéket) a DEBUG írja ki:

I 320
3A

9.2.19 O - output a kijelölt port-ra

Az O (output, adatkivitel) parancs kiírja a kijelölt port-ra (a processzorhoz kapcsolt adatcsatornára) a megadott egy-byte-os értéket. A parancs általános alakja:

O *portcím szám*

ahol a *portcím* egy maximum négyjegyű, a *szám* pedig egy maximum kétjegyű hexadecimális érték lehet.

A példában a 41A0 című port-ra írjuk ki a 2B értéket:
O 41A0 2B

10. Futtatható program előállítása. Programszerkesztés

A programszerkesztő - LINK - végrehajtható programot készít olyan tárgy-file-okból, amelyeket a rendszer különböző szintű fordítóprogramjai készítettek (MASM, C, PASCAL stb.). A szerkesztés eredménye egy program, amelyet a LINK egy végrehajtható program-file-ba (.EXE kiterjesztésűbe) másol. A felhasználó az így készített program futását a file-név PC DOS parancssorban való megadásával indíthatja.

A LINK a megadott tárgy-file-okból, valamint az esetleg szükséges könyvtári file-okból egy relokálható programot készít, amelynek helyigénye legfeljebb 1Mbyte lehet.

A továbbiakban részletesen leírjuk, hogyan vezérelhető a LINK, és hogyan készíti el a futtatható programot.

10.1 A LINK indítása

A LINK indítására és vezérlésére 3 lehetőség van:

- PC DOS parancs-sorral,
- a LINK kulcsszóval, majd a LINK által kiírt kérdésekre adott válaszokkal,
- egy előre összeállított válasz-file-lal.

A LINK működése bármikor leállítható a < Ctrl-Break > lenyomásával.

10.1.1 Programszerkesztés PC DOS parancs-sorral

A szerkesztést indító PC DOS parancs alakja:

LINK [*opciók*]*tárgy-file-ok*,[*futtatható file*],[*map-file*],[*könyvtár file-ok*]

Az "*opciók*" a LINK működését vezérlő feltételek. Ezek részletes leírása e fejezet 10.2. pontjában található.

A "*tárgy-file*" egy vagy több olyan tárgy-file neve, amelyből a programot szerkesztjük. Ezeket a tárgy-file-okat fordítóprogram - pl. makró assembler - készítette.

A "*futtatható file*" a LINK által elkészítendő program-file neve. Ha nem adjuk meg, akkor a LINK az első tárgy-file nevét feltételezi futtatható file-névként .EXE kiterjesztéssel.

A "*map-file*"-ba kerül az elkészült program térképének listája. Ha nem nevezünk meg ilyen file-t, vagy az opciók között nem adjuk meg a /MAP vagy a /LINE opciót, akkor nem készül map-file.

A "*könyvtár-file*" egy vagy több könyvtár-file neve, amely a programhoz szerkesztendő könyvtári rutinokat tartalmazza. Ha ilyenre nincs szükség, akkor a név helyébe pontosvesszőt kell írni.

A különböző típusú file-neveket vesszővel választjuk el egymástól. A vesszők kiírása akkor

is kötelező, ha valamely file nevét nem adjuk meg. Ha egynél több tárgy-file vagy könyvtár-file neve szerepel, akkor a file-neveket legalább egy szóközzel vagy egy " + " jellel kell elválasztani egymástól. Az opciók bárhol lehetnek a parancs-sorban, csak azonos típusú file-nevek között nem.

A parancs-sorban szereplő file-nevekben nem kötelező megadni a kiterjesztéseket, mert a LINK feltételezi rendre a .OBJ (tárgy), .EXE (futtatható), .MAP (térkép) és .LIB (könyvtár) kiterjesztéseket.

Példák:

LINK ujit.OBJ, ujit.EXE, ujit.MAP, ujit.LIB

A parancs hatására a programszerkesztő az UJIT.OBJ tárgyfile-ból az UJIT.EXE végrehajtható program-file-t állítja elő az UJIT.LIB könyvtár-file-ban levő megfelelő rutinok hozzászerkesztésével. A program térképe az UJIT.MAP file-ba kerül.

LINK ujit,ujit,ujit,ujit

A parancs hatása ugyanaz, mint az előzőé, mert a LINK automatikusan hozzáteszi a file-nevekhez a .OBJ, .EXE, .MAP és .LIB kiterjeszést.

LINK ujit,ujit,ujit

A szerkesztés eredménye a LINK alapfeltételezései miatt ugyanaz, mint az előző két esetben.

LINK init + ujit,ujit,ujit,;

Két tárgy-file-ból (init.OBJ és ujit.OBJ) szerkeszti az ujit.EXE file-t könyvtár használata nélkül, és elkészíti az ujit.MAP file-t.

LINK x + y + init/PAUSE,,xy,\kat\elem

Az X.OBJ Y.OBJ és INIT.OBJ tárgy-file-okból, valamint a KAT könyvtárban levő ELEM.LIB alkönyvtár megfelelő rutinjaiból szerkeszti az X.EXE file-t és elkészíti az XY.MAP file-t. A /PAUSE opció hatására a végrehajtás felfüggesztődik, mielőtt a LINK elkezdené a futtatható program előállítását.

10.1.2 Programszerkesztés interaktív üzemmódban

Lehetőség van arra is, hogy a programszerkesztést párbeszédes üzemmódban hajtsuk végre. Ilyenkor az előző pontban ismertetett parancslista elemeit részenként adjuk meg.

A párbeszédes programszerkesztés lépései:

1. Begépeljük a szerkesztő program nevét:

LINK

és megnyomjuk az < ENTER > gombot. Ennek hatására a LINK megkérdezi a tárgymodulok nevét. A képernyőn a következő üzenet jelenik meg:

Object modules [.OBJ]

2. Az előbbi üzenetre válaszul beírjuk az összeszerkesztendő tárgy-file-ok nevét. Egynél több tárgy-file esetén a file-neveket szóközökkel vagy " + " jellel kell elválasztani egymástól. Ha a file-nevek nem férnek el egy sorban, akkor a sor utolsó karaktereként üssünk + jelet, majd nyomjuk meg az < ENTER > -t. Ekkor a LINK megkérdezi a többi tárgy-file nevét. A file-nevekben nem kötelező kiírni a .OBJ kiterjesztést, mert ez az alapfeltételezés.

Ha megadtuk az összes tárgy-file nevét, nyomjuk meg az < ENTER > -t! Ennek hatására a LINK a futtatható file nevét kéri:

Run File [file-név.EXE]

3. Beírjuk a futtatható file nevét és megnyomjuk az < ENTER > -t. Ha nem adunk meg file-nevet vagy kiterjesztést, akkor a LINK az alapértelmezést veszi figyelembe (első tárgymodul neve és .EXE). Ilyenkor elég csak az < ENTER > -t megnyomni.

A LINK ezután a map-file nevét kérdezi:

List File [NUL.MAP]

4. Ha készíteni akarunk map-file-t, akkor írjuk be a nevét és nyomjuk meg az < ENTER > -t! A .MAP kiterjesztést nem kötelező megadni, mert alapfeltételezés. Ha nem kérünk map-file-t, akkor elegendő csak az < ENTER > -t megnyomni.

A LINK következő kérdése a könyvtár neve:

Libraries [.LIB]

5. Valamennyi könyvtár nevét fel kell sorolni, amelyekben olyan rutinok vagy változók vannak, amelyekre a program hivatkozik, de nem deklarálja azokat. A nevek elválasztása szóközökkel vagy + jellel történik. A kiterjesztés alapértelmezése .LIB. A könyvtárneveket több sorban is megadhatjuk úgy, mint a tárgy-file-ok neveit.

Ha könyvtárhasználatra nincs szükség, akkor elég csak az < ENTER > -t leütni.

Ezek után a LINK előállítja a végrehajtható file-t.

Megjegyzés:

1. Ha valamely file másik könyvtárban vagy másik lemezen van, mint az aktuális alapértelmezés, akkor a file nevéhez meg kell adni a megfelelő elérési utat (path), vagy a lemez-azonosítót. Ha a LINK nem talál meg egy tárgy-file-t, akkor üzen és vár, amíg lemezt cserélünk, ha szükséges.
2. Minden sor végén lehetőség van opciók megadására.
3. Bármely üzenetváltás végződhet pontosvesszővel, melynek hatása: a LINK befejezi a párbeszédet, az elhagyott kérdésekre az alapfeltételezést alkalmazza, és elvégzi a programszerkesztést. Ügyeljünk arra, hogy legalább egy tárgy-file neve szerepeljen!

Példa:

A 10.1.1 pontban leírt utolsó példa párbeszédes formája a következő:

```
LINK
Object Modules [.OBJ]: x+y+
Object Modules [.OBJ]: init/PAUSE
Run File [x.EXE]:
List File [NUL.MAP]: xy
Libraries [.LIB]: \kat\elem
```

Az első sort a felhasználó írja be.

A 2. és 3. sor példa a file-nevek többsoros megadására. A 4. sorban láthatjuk a név alapfeltételezését (x), amely a végleges is, mert nem adtunk másikat.

10.1.3 Programszerkesztés előre összeállított válasz-file-lal

A LINK működéséhez szükséges információkat előre elhelyezhetjük egy file-ban, amelyet válasz-file-nak nevezünk. Ekkor a szerkesztés indítása:

LINK @válasz-file-név

A válasz-file-név tetszőleges lehet. A válasz-file tartalma:

tárgy-file-név
[*futtatható file-név*]
[*map-file-név*]
[*könyvtár-file-név*]

A file-ok megadási módjára az előzőekben leírt szabályok érvényesek. Minden file-típusnak külön sorban kell szerepelnie. Ha valamelyik file-típus helyett az alapfeltételezést használjuk, akkor a megfelelő sort üresen kell hagyni.

Indítás után a LINK előveszi a válasz-file-ból a megfelelő információkat, és automatikusan elvégzi a programszerkesztést. Szerkesztés közben a válasz-file tartalma folyamatosan megjelenik a képernyőn. Ha a LINK hibát talál a nevek olvasásakor, hibajelzést ad, és megvárja, amíg beírjuk a helyes választ.

Példa:

Az előző pontban bemutatott párbeszédes szerkesztési példának megfelelő válasz-file a következő:

```
x+y+
init/PAUSE

xy
\kat\elem
```

Figyeljük meg, hogy a harmadik sor nem látszik! Ha az xy ott volna, azt a LINK a RUN File kérdésére adott válasznak tekintené.

10.1.4 Könyvtárak elérési útjainak megadása

A könyvtárak elerési útjainak megadására két módszer lehetséges:

- A LINK parancsban lemezegységek és alkönyvtárak felsorolásával megadhatjuk azokat az elérési utakat, amelyeken haladni kell az adott név keresésekor.
- A második lehetőség az, hogy a LINK hívása előtt a PC DOS PATH parancsával értéket adunk a PATH környezetleíró változónak. Ez az érték egy vagy több elérési út.

Legfeljebb 16 elérési utat adhatunk meg. Az egyes elérési utakat pontosvesszővel választjuk el egymástól.

Egy elérési út leírása:

d:\alkönyvtár

Például:

A:\katall

Ha valamelyik hiányzik, akkor az aktuális lemezegység, illetve alkönyvtár a feltételezés.

Több elérési út megadása esetén a LINK a felírás sorrendjében balról jobbra haladva keresi a megfelelő könyvtárat.

Ha például a kt.LIB könyvtár elérési útjai:

A:\kat1;\kat2;B:

akkor először az "A" lemezegység \KAT1 alkönyvtárában keresi KT.LIB-et. Ha nem találja, az aktuális lemezegységen a \KAT2 alkönyvtár következik, s ha abban sincs, áttér a B: lemezegységre és az aktuális alkönyvtárban folytatja a keresést.

Példák:

1. LINK rend,,rend.MAP,A:\kat1\kt.LIB + dcl + B: + C:\kat2\

A LINK a KT.LIB könyvtárat csak az A: lemezegység KAT1 alkönyvtárában keresi, mert közvetlenül a könyvtárnév előtt adtuk meg az elérési utat. A DCL könyvtárat először az A: lemezegység aktuális könyvtárában keresi, majd a B: lemezegység aktuális alkönyvtárában, végül a C lemezegyseg KAT2 könyvtárában.

2. SET PATH = A:\kat1;B:;C:\kat2
   LINK rend,,rend.MAP, kt + dcl

Ebben a példában a LINK először az aktuális lemezegység aktuális alkönyvtárában keresi a KT és a DCL könyvtárat. Ha nem találja, akkor az A: egység KAT1 alkönyvtárában, a B: egység aktuális alkönyvtárában, majd a C: egység KAT2 alkönyvtárában folytatja a keresést.

10.1.5 A map-file szerkezete

A programról készített map-file lényegében a program főbb részeinek a program kezdetéhez relatív elhelyezkedéséről ad információt, valamint jelzi az esetleges hibákat.

Ha a LINK parancsban megadtuk a /MAP opciót, akkor kilistázza a public szimbólumokat és azok betöltési címeit is.

A map-file-ban levő szegmens-információ alakja:

START	STOP	LENGTH	NAME	CLASS
kkkkkH	vvvvvH	llllllH	x...x	y...y
.	.	.	.	
.	.	.	.	
.	.	.	.	

A táblázat első két oszlopa a szegmensek kezdő és végcímét mutatja 20 bites hexadecimális alakban. Ezek a címek a load modul kezdetéhez relativek.

A load modul feltételezett kezdőcíme 00000H. Ezekből a címekből képezi az operációs rendszer a valódi címeket a program betöltésekor.

A harmadik oszlop a szegmens byte-okban mért hosszát adja meg.

A negyedik oszlopban van a szegmens neve, az ötödikben a szegmens típusa (pl. CODE utasítások, DATA adatok).

A map-file-ban levő csoport-információ általános alakja:

ORIGIN	GROUP
nnnn:n	x...x
.	.
.	.
.	.

Az első oszlop a csoport kezdetét, a második a nevét mutatja.

A LINK a program belépési pontjának címét a map-file végén helyezi el.

Ha a LINK parancssorában megadjuk a /MAP opciót, akkor a LINK a public szimbólumokról is elhelyez két táblázatot a map-file-ban. Az egyik táblázatban abc sorrendben, a másikban betöltési cím szerinti sorrendben szerepelnek a szimbólumok. A címek szegmens: relatív cím alakban kerülnek a map-file-ba. Ezek a load-modul kezdetéhez relatív címek.

Ha a LINK parancsban a /HIGH és /DSALLOCATION opció is szerepel és a program mérete az adatokkal együtt nem haladja meg a 64 Kbyte-ot, akkor az előzőekben említett táblázatokban az egyes szimbólumok mellett meglepően nagy címet találhatunk. Ez azt jelenti, hogy a szimbólum a program elé kerül.

10.1.6 VM.TMP - Ideiglenes lemez-file

A LINK munkaterülete a tárban van. Ha ez a terület programszerkesztés közben megtelik, akkor a LINK létrehoz egy ideiglenes lemez-file-t, amelynek neve az aktuális alkönyvtárban VM.TMP lesz. Ekkor a LINK a következő üzenetet küldi:

VM.TMP has been created.
Do not change diskette in drive <d:>

Az üzenet jelentése:

VM.TMP készült.
Ne cserélje a lemezt a <d> egységben!

Ez az üzenet a LINK működése végéig érvényes. A szerkesztés végrehajtása után az ideiglenes lemez-file automatikusan törlődik.

Megjegyzés:

Ha a LINK ideiglenes file-t hoz létre, az esetleg létező VM.TMP nevű file-t törli. Ezért ezt a nevet saját célra ne használjuk!

10.2 A LINK opciói

Az opciók segítségével vezérelhetjük a LINK működését. Minden opció "/" jellel kezdődik és a LINK parancsban bárhol előfordulhat, csak azonos típusú file-nevek között nem. A gyorsabb írás kedvéért minden opció neve rövidíthető.

Az opciók listája:

/PAUSE	megszakítja a szerkesztést
/MAP	kiírja a public szimbólumok listáját
/STACK	verem-méret
/HIGH	betöltés a lehetséges legnagyobb címtől
/DSALLOCATION	helyfoglalás adatcsoportnak
/LINE	forrás-sorok kezdőcímének kiírása
/X	az .EXE file által tartalmazott szegmensek száma
/O	a PASCAL és a FORTRAN fordítók 1. verziója által fordított modulok

10.2.1 PAUSE - Megszakítás a szerkesztés alatt

Alakja:

/PAUSE rövidítve: **/P**

Ezzel az opcióval megszakítható a LINK működése, mielőtt a végrehajtható file-t (.EXE) lemezre írja. Akkor szükséges, ha a kiírás előtt lemezt akarunk cserélni.

Ekkor a LINK üzenete:

About to generate .EXE file
Change disks < hit ENTER >

Az üzenet jelentése: Az .EXE file generálásához cserélje ki a lemez(eke)t és nyomja meg az ENTER billentyűt!

Az < ENTER > billentyű lenyomása után a LINK folytatja a munkát.

Ha VM.TMP nevű file van már a lemezen, akkor ezt a lemezt nem szabad cserélni.

Példa:

LINK rend/PAUSE,,,\kat\elem

A parancs hatása az, hogy a rend.EXE file lemezre írása előtt a LINK megáll, majd a lemezcsere után folytatja működését. A rend.EXE file lemezre írása után ismét megáll, lehetővé téve az eredeti lemez visszahelyezését.

10.2.2 MAP - Public szimbólumok táblájának előállítása

Alakja:

/MAP rövidítve: **/M**

Az opció hatására a LINK elkészíti a public szimbólumok táblázatát abc sorrendben, valamint cím szerint növekvő sorrendben (ld.10.1.5) és elhelyezi a map-file-ban.

Ha a LINK parancsban nem jelöltünk ki map-file-t, akkor a lista-file előtt megadott /MAP opció hatására automatikusan készül ilyen file, amelynek neve az első tárgy-file neve .MAP kiterjesztéssel.

Példa:

LINK rend/MAP;

ami egyenértékű a

LINK rend.OBJ,rend.EXE,rend.MAP/MAP,;

paranccsal.

10.2.3 STACK - A verem-méret meghatározása

Alakja:

/STACK:*méret* rövidítve: **/S:***méret*

Hatására a programhoz tartozó verem helyfoglalása "*méret*" byte lesz. Értéke

1< = méret < = 65535

lehet.

A méret decimális formában adható meg.

A LINK általában automatikusan kiszámítja a verem méretét a tárgymodulokban megadott verem-szegmensek méretei alapján. Ezt a számított értéket lehet felülbírálni a LINK parancs /S opciójával.

Ha nincs verem-szegmens a programunkban, akkor a LINK hibajelzést ad. Ezt elkerülendő, minden programban legalább egy vermet definiáljunk!

Megjegyzés:

Amennyiben programokból .COM file-t akarunk előállítani, nem szabad STACK szegmenst definiálnunk. Ilyenkor figyelmen kívül hagyhatjuk a programszerkesztő figyelmeztető üzenetét. (A .COM előállítására .EXE file-ból az EXE2BIN segédprogramot használhatjuk.)

Példa:

```
LINK rend/S:512;
```

vagy más alakban

```
LINK rend.OBJ/STACK:512,rend.EXE;
```

Mindkét parancs 512 byte helyet biztosít a verem számára a REND nevű programban (futtatáskor).

10.2.4 HIGH - A kezdőcím maximumra állítása

Az opció hatására a program kezdőcíme a lehető legnagyobb cím lesz a tár szabad területén.

Az opció megadási módja:

/HIGH rövidítve: **/H**

Így lehet a programot a szabad hely végében elhelyezni.

Ha nem adunk meg ilyen opciót, akkor a betöltés a szabad hely elején kezdődik.

Csak a /DSALLOCATION paraméterrel együtt használjuk.

10.2.5 DSALLOCATION - Helyfoglalás az adatcsoportnak

Az opció hatására a DGROUP nevű csoportot tartalmazó szegmensben a csoport a lehető legnagyobb címre kerül.

Az opció alakja:

/DSALLOCATION rövidítve: **/D**

A LINK alapértelmezésben az adatcsoport első byte-jának szegmens-relatív címét 0000H-nak adja meg, míg az opció hatására a csoport utolsó byte-jához az FFFFH relatív címet rendeli.

Ezt az opciót legtöbbször a /HIGH opcióval együtt használjuk, mert így biztosítani tudjuk, hogy a program előtti fel nem használt tárterülettel jobban gazdálkodjunk.

10.2.6 LINE - Forrás-sorok kezdőcímének kiírása

Az opció hatására a LINK program minden forrás-sorából képzett utasítás(ok) kezdőcímét a map-file-ba írja.

Az opció alakja:

/LINE rövidítve: **/L**

A LINK által így készített információt fel lehet használni magasabb szintű nyelven írt programok forrásszintű nyomkövetéssel való teszteléséhez.

Az opció megadásának hatására csak akkor képez a LINK utasítás-kezdőcímeket, ha:

- van map-file-név a LINK parancsban,
- a tárgy-file-ban van sorszámozás (csak a magasszintű nyelvek fordítói készítenek ilyet).

Ha nincs a tárgy file-ban sorszámozás, akkor a LINK figyelmen kívül hagyja a LINE opciót.

Megjegyzés:

Ha nem adtunk meg map-file-nevet,akkor a LINE a MAP opció funkcióját is ellátja.

10.2.7 X - Szegmensszám beállítása

Az opcióval megadhatjuk az .EXE file által tartalmazott szegmensek maximális számát. Ez az érték 1-1024 között változhat. Ha az egyik opciót nem használjuk, akkor a programszerkesztő a 256 értékkel dolgozik.

Az opció alakja:

/X:*n* ahol *n*: 0-1024 lehet

10.2.8 O - 'Régi' fordítók használata

Az O (old) opció lehetővé teszi a Microsoft Pascal és FORTRAN fordítók 1. verziójával fordított file-ok szerkesztését is.

Az opció alakja: **/O**

11. A rendszer konfigurálása

A DOS az indítás során annak a meghajtónak a főtartalomjegyzékében, ahonnan indítottuk, megkeresi a CONFIG.SYS nevű konfigurációs file-t. A file-ban speciális parancsok vannak. Ezeket a DOS értelmezi, és ezek alapján beállítja a rendszer működésének jellemzőit.

A konfigurációs file tartalmának meghatározásakor figyelembe kell venni:

- az adott számítógép konfigurációját (memória mérete, fix és floppy meghajtók száma stb.),
- az adott számítógépen milyen alkalmazó rendszerek futnak (pl. dBASE III, sok I/O művelet),
- az is lehetséges, hogy egy-egy alkalmazói rendszer futtatása előtt a DOS rendszert újraindítjuk egy új konfigurációs file-lal.

11.1 A konfigurációs file kezelése

A felhasználó új CONFIG.SYS file-t szerkesztőprogrammal (pl. EDLIN) vagy a COPY paranccsal hozhat létre. Utóbbi esetben a következők a teendők. A DOS prompt-ja után begépeljük:

A > COPY CON CONFIG.SYS

és < ENTER > -t adunk, majd beírjuk a konfigurációs parancsokat, mindegyik után < ENTER > -t ütve. A COPY CON parancsot az < F6 >, majd az < ENTER > billentyűvel zárjuk, ennek hatására a file-t a rendszer megőrzi.

Az új file hatása csak a következő rendszerindítás után érvényesül.

11.2 A konfigurációs parancsok

11.2.1 BREAK - a Break billentyű figyelése

Verzió: 2.11 - 3.30
A parancs formája:

BREAK [ON|OFF]

A paraméterek jelentése:

A rendszer az **ON** megadása esetén a DOS minden funkciójának végrehajtása előtt ellenőrzi, hogy a < Ctrl-Break > billentyűt leütöttük-e.

Az **OFF** megadása esetén csak konzol I/O, nyomtatási vagy aszinkron kommunikációs adapteri (soros átviteli) művelet esetén ellenőrzi a < Ctrl-Break > billentyűt.

Megjegyzés:

1. A CONFIG.SYS file-ban megadott BREAK-kapcsoló állapotát a BREAK DOS paranccsal meg lehet változtatni.

2. Ha a CONFIG.SYS file-ban nem adunk meg BREAK parancsot, akkor az a BREAK = OFF magadással azonos.

11.2.2 BUFFERS - lemez pufferek száma

Verzió: 2.11 - 3.30
A parancs formája:

BUFFERS = *xx*

A paraméter jelentése: Az *xx* 1-99 közötti egész szám, amely a lemez-pufferek számát jelenti. Minden egyes puffer a DOS méretét 528 byte-tal növeli, ezért a felhasználói memóriaterület csökken.

Alapértelmezés: 2 puffer (IBM PC AT esetén 3 puffer)

Megjegyzés:

1. A pufferek optimális számát a következő információk alapján állapíthatja meg a felhasználó:
   - a leggyakrabban használt alkalmazások típusa,
   - a számítógép memóriájának mérete,
   - a rendszer működésének elemzése különböző puffer-számok esetén.
2. A pufferek száma fix lemez esetén legalább 3 legyen.
3. Legalább 512 Kbyte memóriával rendelkező gépen javasolt legalább 20 puffert megadni: BUFFERS = 20

11.2.3 COUNTRY - dátum, idő és egyéb formátum megadása

Verzió: 3.00 - 3.30
A parancs formája (DOS 3.00-3.20):

COUNTRY = *xxx*

DOS 3.30 esetén a COUNTRY alakja a következő:

COUNTRY = *xxx*[,*yyy*],[*d:*]*file-név*[.*kit*]
COUNTRY = *xxx*,[*yyy*]

A paraméter jelentése:

Az *xxx* 3 decimális jegyből álló szám, az adott ország telefon kódja. Kezdeti értéke 001, ez az US formátum beállításának felel meg.

A konfigurációs parancs a dátum és idő kijelzési formátumát, a pénznem jelét és a tizedes helyét kijelölő karaktert állítja be. A DOS 3.20 az alábbi országokat ismeri fel:

Ausztrália	061
Belgium	032

Kanada (francia)	002
Dánia	045
Finnország	358
Franciaország	033
NSZK	049
Olaszország	039
Izrael	972
Közép Kelet	785
Hollandia	031
Norvégia	047
Portugália	351
Spanyolország	034
Svédország	046
Svájc	041
Egyesült Királyság	044
Egyesült Államok	001

yyy az adott ország kódhoz tartozó két kódlap valamenyike.

[*d:*]*file-név*[*.kit*] az országspecifikus információk használati módjára (pl.dátum, idő kijelzése) utaló információt tartalmazza. Ha elhagyjuk, akkor a program a COUNTRY.SYS file-lal dolgozik.

11.2.4 DEVICE - periféria-kezelő rutinok

Verzió: 2.11 - 3.20
A parancs formája:

DEVICE = [*d:*][*út*]*file-név*[*.kit*][*paraméterek*]

A paranccsal egy periféria-kezelő rutint tartalmazó file-t adunk meg. A DOS rendszer beépített periféria-kezelő rutinjai a következő egységek kezelését végzik:

- képernyő,
- billentyűzet,
- nyomtató,
- floppy egység (160/180 Kbyte, 320/360 Kbyte, 1.2 Mbyte),
- fix lemez.

Ha a számítógép konfigurációja csak ezeket a perifériákat tartalmazza, akkor nem kell megadni a DEVICE konfigurációs parancsot.

A DOS 3.30 változata öt periféria-kezelő rutint használ. Ezek a következők:

ANSI.SYS
DISPLAY.SYS
DRIVER.SYS
PRINTER.SYS
VDISK.SYS

Az ANSI.SYS az ANSI szabvány szerinti képernyőkezelést biztosítja. A VDISK.SYS me-

mórialemezt hoz létre, csak a DOS 3.00 vagy későbbi változataihoz használható. A DRIVER.SYS fizikai és logikai lemezegységek egymáshoz rendelését végzi el, s lehetővé teszi külső fizikai egységek használatát is. Csak a DOS 3.20-tól használható! A DOS 3.30-as változatától a nyomtatóra és a képernyőre biztosítja a kód-lapozás használatát a PRINTER.SYS, illetve a KEYB.SYS periféria-kezelő. A továbbiakban részletesen ismertetjük ezek használatát.

11.2.4.1 A kibővített képernyő- és billentyűzet-kezelés (ANSI.SYS)

A 2.00 az első olyan DOS verzió, amely megengedi, hogy a felhasználói programok speciális jelsorozatok kiadásával (escape-szekvencia) vezérelhetik a képernyőt és a billentyűzetet.

Megjegyzés:

E lehetőség alkalmazásához szükséges, hogy a kibővített képernyő- és billentyűzet-kezelés a DOS rezidens részét alkossa.

Ez úgy érhető el, hogy a CONFIG.SYS file-ba beszerkesztjük (pl. EDLIN-nel) a következő utasítást:

DEVICE = ANSI.SYS

Ez behívja a memóriába - a rendszer újraindításakor - a szükséges kezelőprogramot. Arra figyelemmel kell lenni, hogy a DOS memóriaigénye az ANSI.SYS méretével megnövekszik.

A vezérlő parancsok formája:

ESC[*paraméterek feladatazonosító*

Figyelem! A [jel része a parancsnak, közvetlenül az ESC (decimális 27, hexadecimális 1B) után áll!

A paraméterek és a feladatazonosítók lehetséges értékeit a további alszakaszokban ismertetjük.

A fenti "parancs" nem DOS parancs az eddigi értelemben, nem adhatjuk ki a fent leírt formában, a DOS akkor szintaktikus hibát jelezne (Bad command or filename).

Az egész "parancssort" az ESC ASCII kódjának (hexadecimális 1B, decimális 27) figyelembevételével kell a képernyőre kiírni, vagy a DEBUG, vagy a BASIC segítségével.

```
e200                                1B "[2;10H"
az adatszegmens 200-as              ezt az escape szekvenciát
címére beteszi a DEBUG              (itt a 2;10H a paraméter és feladatazonosító)
```

Ezután, ha ezt a címet a képernyőre küldjük, a kurzor a 2. sor 10-ik oszlopára áll. Ugyanezt BASIC-ből:

```
10 PRINT CHR$(27); "[2;10H"
```

Mint látjuk, a paramétereknek és a feladatazonosítónak egy összefüggő sztringben kell lennie.

A "paraméterek" numerikus értékek. Az egyes lehetőségek leírásánál annyi kettőskeresztet (#) fogunk megadni, ahány ilyen érték az adott lehetőségnél megadható. A numerikus értékek decimális pozitív egész számok.

Ha egy paramétert elhagyunk, vagy 0-nak adunk meg, a paraméter hallgatólagos értéke lesz érvényes.

A "feladatazonosító" az adott lehetőséget azonosítja.

A kibővített képernyő-kezelés

- a kurzor mozgatását,
- a képernyő törlését és
- üzemmódjának beállítását végzi.

1. A kurzor állítása: ESC[#;#H

A kurzort a paraméterekkel megadott pozícióra mozgatja. Az első érték a sor, a második az oszlop értékét adja. Az alapértelmezés 1;1. Ha paramétert nem adunk meg, akkor a kurzor alaphelyzetbe (HOME) áll.

2. Kurzor fel: ESC[#A

A # azt adja meg, hogy a kurzort hány sorral mozgassa felfelé. Alapértelmezés:1. A képernyő tartalma mozgatás közben nem változik. Ha a kurzor a legfelső sorban áll, a parancs hatástalan.

3. Kurzor le: ESC[#B

A # azt adja meg, hogy a kurzort hány sorral mozgassa lefelé. Alapértelmezés:1. A képernyő tartalma mozgatás közben nem változik. Ha a kurzor a legalsó sorban áll, a parancs hatástalan.

4. Kurzor előre (jobbra): ESC[#C

A # azt adja meg, hogy a kurzort hány oszloppal mozgassa jobbra. Alapértelmezés:1. A képernyő tartalma mozgatás közben nem változik. Ha a kurzor a jobbszélső oszlopban van, a parancs hatástalan.

5. Kurzor vissza (balra): ESC[#D

A # azt adja meg, hogy a kurzort hány oszloppal mozgassa balra. Alapértelmezés: 1. Mozgatás közben a képernyő tartalma nem változik. Ha a kurzor a balszélső oszlopban van, a parancs hatástalan.

6. Kurzor vízszintes és függőleges mozgatása: ESC[#;#f

A kurzort a paraméterekkel megadott pozícióra mozgatja. Az első a sor, a második az oszlop sorszámát adja meg. Ha nem adunk meg értéket, akkor a kurzor alaphelyzetbe (HOME) áll.

7. Kurzor helyzetének kijelzése: ESC[6n

Ez lehetővé teszi a kurzor helyzetének folyamatos kijelzését.

8. Kurzor pillanatnyi helyzetének kijelzése a standard input periférián: ESC[#;#R

A kurzor pillanatnyi helyzetét adja meg. a standard input periférián. Az első paraméter a pillanatnyi sor, a második az oszlop sorszámát adja meg.

9. Kurzor pozíciójának megőrzése: ESC[s

A kurzor pillanatnyi helyzetét a rendszer megjegyzi. Ezt a pozíciót a 10. paranccsal lehet visszaállítani.

10. Kurzor pozíciójának visszaállítása: ESC[u

A kurzort abba a pozícióba helyezi, amelyben ESC[s kiadásakor volt.

11. Képernyő törlése: ESC[2J

A teljes képernyő törlődik, a kurzor alapállapotba kerül.

12. Törlés sor végéig: ESC[K

A kurzor pozíciójától a sor végéig valamennyi jelet töröl, azt is, amelyen a kurzor áll.

13. A grafikus attributumok beállítása: ESC[#;...;#m

értékei a következők lehetnek:

0 normál fekete-fehér kijelzés
1 dupla erősségű kijelzés (high intensity)
4 aláhúzás bekapcsolása (csak IBM monokróm képernyő esetén használható)
5 villogó kijelzés (blink on)
7 fordított kijelzés (reverse video on)
8 törlés bekapcsolása (a képernyőn nem látható)
30 az előter színe fekete
31 az előtér színe piros
32 az előtér színe zöld
33 az előtér színe sárga
34 az előtér színe kék
35 az előtér színe viola
36 az előtér színe kobaltkék
37 az előtér színe fehér
40 a háttér színe fekete
41 a háttér színe piros
42 a háttér színe zöld
43 a háttér színe sárga
44 a háttér színe kék
45 a háttér színe viola
46 a háttér színe kobaltkék
47 a háttér színe fehér.

14. Képernyő beállítása: ESC[= #h

értéke a következő lehet:

0 40x25 fekete-fehér
1 40x25 színes
2 80x25 fekete-fehér
3 80x25 színes
4 320x200 színes grafikus
5 320x200 fekete-fehér grafikus
6 640x200 fekete-fehér grafikus
7 A "sor vége" beadása után (END OF LINE) új sort kezd (Line Feed).

15. Képernyő visszaállítása: ESC[= #I

értéke ugyanaz lehet, mint a 14. pontnál leírtak. Ha viszont az ESC[= 7I értéket adjuk meg, akkor a "sor vége" beadása után (END OF LINE) nem lép új sorba.

A DOS lehetőséget biztosít arra, hogy a tasztatúrán lévő jel-billentyűk jelentését újra definiáljuk, vagy egy billentyűhöz karaktersorozatot rendeljünk.

1. ESC[#;#;...;#p

A # egy ASCII kód. Az első # az átalakítandó kód, a többi pedig azoknak a jeleknek az ASCII kódja, amelyek akkor jelennek meg a képernyőn, ha a megadott billentyűt lenyomjuk.

2. ESC[#;"karaktersorozat";#p

Az első # által meghatározott ASCII kódhoz egy karaktersorozatot rendelünk.

Megjegyzés:

Ha az első # értéke 0 (nulla), akkor ezt még egy # követi. A két # egy kibővített ASCII kódot jelöl.

Példa:

1. A kis q billentyűhöz a nagy Q-t rendeljük hozzá:

ESC 113;81p

2. A < PF10 > billentyűhöz a DIR karaktersorozatot rendeljük, majd ezt a "sor vége" kóddal le is zárjuk.

ESC 0;68;"DIR";13p

A 0;68 az < PF10 > funkció-billentyűt jelenti, a 13 pedig a "sor vége"-kód.

11.2.4.2 DISPLAY.SYS - kódlapok használatának engedélyezése a képernyőn

A periféria-kezelő használata lehetővé teszi, hogy azokon az eszközökön, amelyekerre fizikailag alkalmasak, különböző kódlapokat használjunk. Ezek az eszközök: IBM LCD tí-

pusú képernyő, EGA grafikus adapterű gépek és az IBM PS/2 típusú mikroszámítógépek.

Verzió: 3.30
A parancs formája:

1. **DEVICE** = [*d:*][*út*]**DISPLAY.SYS CON**[:] = (*típus*[,[*kp*][,*n*]])
2. **DEVICE** = [*d:*][*út*]**DISPLAY.SYS CON**[:] = (*típus*[,[*kp*][,(*n*,*m*)]])

A paraméterek jelentése:

típus a rendszerben levő grafikus kártya típusát adja meg. Lehetséges értéke: **MONO**, **CGA**, **EGA** vagy **LCD**. Ha IBM PS/2 típusú gépünk van, akkor is az EGA-t kell megadnunk, holott a két grafikus adapter funkcionálisan nem ekvivalens. Ilyen esetben a program automatikusan felismeri, hogy melyikről van szó.

kp a szóbanforgó hardware által támogatott kódlap sorszáma. A szóbajöhető értékek: **437**, **850**, **860**, **863** és **865**.

n a kiegészítő kódlapok számát adja meg, n értéke a 0-12 tartományba kell, hogy essen. MONO és CGA típus esetén n értéke értelemszerűen 0.

m a betűtípusok számát adja meg az adott képernyő-típuson belül. Ha m-et nem adjuk meg, akkor azt a típus függvényében választja meg a program. LCD esetén (karakterméret 8x8) m = 1. EGA (karakterméret 8x8 vagy 8x14) és IBM PS/2 (karakterméret 8x8 vagy 8x16) esetén m = 2

út a DISPLAY.SYS file-t tartalmazó tartalomjegyzék.

Példa:

DEVICE = C:\DOS\DISPLAY.SYS CON: = (EGA,865,2)

11.2.4.3 A DRIVER.SYS használata

A DEVICE konfigurációs parancsban használhatjuk a DRIVER.SYS periféria-kezelő rutint, mely lehetővé teszi fizikai és logikai lemezegységek egymáshoz rendelését.

Verzió: 3.20-3.30
A parancs formája:

DEVICE = [*d:*][*út*]**DRIVER.SYS /D:***ddd*[**/T:***ttt*][**/S:***ss*][**/H:***hh*][**/C**][**/N**][**/F:***f*]

A paraméterek jelentése:

/D *ddd* a fizikai meghajtó számát adja. A fizikai meghajtó szám a 0-255 intervallumba eshet. A 0-ás az első belső, az 1-es a második belső hajlékonylemezes meghajtó. A 2-es meghajtó szám a harmadik hajlékonylemezes egységé, amelyik azonban már csak külső egység lehet. Az első fix lemezegység fizikai meghajtó száma 128, a másodiké 129 stb.

/T *ttt* megadja az egy oldalon levő sávok (track) számát. Kezdeti értéke 80 sáv/oldal. (Lehetséges érték: 1-999.)

/S *ss* megadja az egy sávon levő szektorok számát. Kezdeti értéke 9 szektor/sáv. (Lehetséges érték: 1-99.)

/H *hh* a lemez író/olvasó fejeinek maximális száma. Kezdeti érték: 2. (Lehetséges érték: 1-99.)

/C jelzi, hogy a lemezegység kezeli az ún. "changeline" opciót (érzékeli hogy kicserélték a lemezt). Ennek csak IBM PC AT és kompatibilis gépek esetén van hatása.

/N megadja, hogy az egység nem cserélhető. Ilyenek a fix lemezek.

/F *f* megadja a periféria típusát. f lehetséges értékeit az alábbi táblázat tartalmazza:

f	egység
0	160/180 Kbyte
0	320/360 Kbyte
1	1.2 Mbyte
2	720 Kbyte
7	1.44 Mbyte

Kezdeti értéke 2.

Példa:

1. Gépünk (egy IBM PC XT) egy belső hajlékonylemezt és egy fix lemezt tartalmaz. Ha a géphez csatlakoztatott (külső) 720 Kbyte-os egységet szeretnénk használni, akkor a CONFIG.SYS file-ba az alábbi parancsot kell beírnunk:

```
DEVICE = DRIVER.SYS /D:2
```

Ebben az esetben a külső lemezegységet a D betűjellel használhatjuk. Ha ezt az egységet másolásra is akarjuk használni, akkor a parancsot kétszer kell a CONFIG.SYS file-ban elhelyezni:

```
DEVICE = DRIVER.SYS /D:2
DEVICE = DRIVER.SYS /D:2
```

Az első parancsra létrejön egy D, a másodikra egy E betűjelű lemezegység. Ezután kiadhatjuk pl. a DISKCOPY E: D: parancsot. Fizikailag mindkettő a harmadik (külső) hajlékonylemezes egységgel azonos.

Előfordul, hogy a második belső lemezegység (B:) helyére helyezik el a streamert. Ebben az esetben a DISKCOPY A: B: parancs értelemszerűen nem használható. Ebben az esetben ehelyett létrehozunk egy D: logikai egységet, amelyik ugyancsak a 0-ás belső fizikai egységet használja:

```
DEVICE = DRIVER.SYS /D:0 /T:40 /S:15 /H:2 /F:0
```

Ennek hatására a DISKCOPY A: D: parancsot használhatjuk.

11.2.4.4 PRINTER.SYS - kódlapok használatának engedélyezése a nyomtatásra

A periféria-kezelő lehetővé teszi, hogy azokon a nyomtatókon, amelyek erre fizikailag alkalmasak, különböző kódlapokat használjunk. Jelenleg két nyomtató típust támogat a periféria-kezelő, az IBM Proprinter Model 4201 és az IBM Quitwriter III Printer Model 5202 típusokat.

Verzió: 3.30

A parancs formája:

1. **DEVICE** = [*d:*][*út*]**PRINTER.SYS LPT#**[:] = (*típus*[,[*kp*][,*n*]])
2. **DEVICE** = [*d:*][*út*]**PRINTER.SYS LPT#**[:] = (*típus*[,[(*kp1*,*kp2*,...)][,*n*]])

A paraméterek jelentése:

út	a PRINTER.SYS file-t tartalmazó tartalomjegyzék.
#	a nyomtató sorszáma. A használható eszköznevek: LPT1, LPT2 és LPT3. Ennek megfelelően # értéke **1**, **2** vagy **3** lehet. LPT1helyett írhatunk PRN-t is.
típus	a szóbanforgó nyomtató típusa: 4201 vagy 5202.
kp,kp1,kp2...	a használni kívánt, a hardware által támogatott kódlapok száma. Lehetséges értékei: **437**, **850**, **860**, **863**, **865**.
n	a még használható kódlapok száma. Maximális érték 12.

Példa:

DEVICE = PRINTER.SYS LPT1: = (4201,850,)

11.2.4.5 VDISK.SYS - memórialemez használata

A VDISK.SYS periféria-kezelővel memória lemezt hozhatunk létre. A memória lemez pontosan úgy használható, mint a rendes lemez, azzal a lényeges különbséggel, hogy a gép kikapcsolásakor ennek adatai elvesznek. Memórialemez létrehozásához az alábbi parancsot kell elhelyezni a CONFIG.SYS file-ban:

DEVICE = [*d:*][*út*]**VDISK.SYS**[*megj*][*bbb*][*megj*][*sss*][*megj*][*ddd*][/**E**[:*m*]]

Verzió: 3.00 - 3.30

A paraméterek jelentése:

[*d:*][*út*]	azt a tartalomjegyzéket adja meg, ahol a VDISK.SYS file található.
megj	tetszőleges karaktersorozat, melyet megjegyzésekre lehet használni.
[*bbb*]	megadja a memória lemez nagyságát Kbyte-okban. Kezdeti értéke 64 Kbyte. A periféria-kezelő ezt az értéket az alábbi szabályok szerint módosíthatja:
a)	Ha a memórialemez installálása után kevesebb, mint 64 K memória maradna, a rutin csökkenti a bbb értékét. Erről üzenetet kapunk.
b)	Ha a memóriában már eleve 64 K-nál kisebb hely van, akkor a rutin nem installálja a memória lemezt. Erről is üzenetet kapunk.
[*sss*]	a szektor méret byte-okban. A lehetséges értékek: 128, 256 vagy 512. Ha nem adjuk meg, vagy helytelen értéket adunk meg, akkor a rutin a 128 értéket használja.
[*ddd*]	a főtartalomjegyzék bejegyzéseinek száma. (2 és 512 között lehet)
[/**E**[:*m*]]	megadja, hogy a memória lemezt kibővített memória területen használjuk-e. Ezt az opciót természetesen csak akkor használhatjuk, ha a gépben van kibővített memória (1Mbyte felett). m adja meg, hogy egyszerre maximum hány szektort lehet átvinni a bővített memória és a memória között. m lehetséges értéke 1,2,3,4,5,6,7 és 8. Kezdeti érték 8.

Megjegyzés:

1. A főtartalomjegyzék bejegyzések számát is módosítja a rutin, úgy, hogy a legközelebbi szektorhatárig további bejegyzéseket foglal le.
2. Ha a memória lemez nagysága kevés ahhoz, hogy a "boot" szektort, a FAT-ot és a főtartalomjegyzéket és még két szektort tartalmazzon, akkor a rutin ennek megfelelően megnöveli a memória lemez méretét.
3. A memória lemeznek automatikusan generál a rutin egy kötetazonosítót.
4. A memória lemez nem formázható meg.
5. A memória lemezek számának és méretének csak a rendelkezésre álló memóriaterület szab határt.
6. A /E paraméter esetén a 19H megszakítást használja a rutin a bővített memória kezelésére.

Példa:

Tegyük fel, hogy gépünk egy belső hajlékonylemezes és egy fixlemezes egységet tartalmaz. A

DEVICE = VDISK.SYS 180 512 64

konfigurációs parancs segítségével egy 180 Kbyte nagyságú memória lemezt hozunk létre. Szektorméret 512, a főtartalomjegyzék bejegyzéseinek száma 64. A memória lemezre a D betűjellel hivatkozhatunk.

11.2.5 FCBS - maximálisan megnyitható FCB-k száma

A konfigurációs parancs segítségével megadhatjuk, hogy egyidejűleg maximum hány file kontroll blokkot (FCB-t) nyithat meg a DOS.

Verzió: 3.00 - 3.30
A parancs formája:

FCBS = *m*,*n*

A paraméterek jelentése:

m az egyidejűleg megnyitható FCB-k száma. Értéke 1 és 255 között változhat. Alapértelmezése: 4.

n azon FCB-k száma, amelyeket nem lehet automatikusan lezárni, amikor a DOS az m + 1-ik FCB-t kisérli megnyitni. A DOS, ha az m + 1-ik FCB-t akarjuk megnyitni, kénytelen valamelyik előző megnyitottat lezárni. n megadása esetén az első n megnyitott FCB nyitva is marad. Lehetséges értéke: 0-255. Kezdeti érték:0.

Megjegyzés:

1. m legalább akkora kell, hogy legyen, mint n.

2. Ha egy program több FCB-t használ ugyanarra a file-ra, azt a DOS csak egynek tekinti.

3. Az FCBS parancsnak csak akkor van hatása, ha a file megosztási környezetet a SHARE DOS paranccsal betöltöttük. Ha az m+1-ik FCB-t akarjuk megnyitni, akkor a legrégebben használt FCB-t (de nem az első n közül!) lezárja a DOS. Előtte azonban

   FCB unavailable
   Abort? Retry? Ignore?

   üzenetet kapunk.

11.2.6 FILES - egyidejűleg megnyitható file-ok száma

Verzió: 3.00 - 3.30

A parancs formája:

FILES = xxx

A paraméter jelentése:

A FILES = xxx az egyidőben megnyitott file-ok maximális számát jelenti.

Az xxx 8-255 közötti egész szám. A DOS a file-ok kezelése során minden file-hoz egy file-leíró blokkot rendel. A DOS 8 db file-leíró blokkot a saját területén tud tárolni, tehát az alapértelmezés:8. Ha nem elégséges ez a terület, akkor a FILES érték megnövelésével lehet pótlólagos memória területet lefoglalni.

Megjegyzés:

1. Ha nem elégséges a terület a file-ok megnyitásához, akkor hibaüzenetet kapunk.

2. A FILES = xxx érték meghatározásánál nemcsak a felhasználói programok, hanem a rendszer-komponensek file-jait is figyelembe kell venni, Így pl. dBASE III esetén a FILES értéke legalább 24 legyen.

11.2.7 LASTDRIVE - a legnagyobb lemezegység megadása

A konfigurációs parancs megadja a logikailag használható lemezegységek maximális számát.

Verzió: 3.00 - 3.30

A parancs formája:

LASTDRIVE = *d*

A paraméter jelentése:

d az utolsó, még logikailag használható lemezegység betűjele. Ha pl. 14 egységet akarunk használni, akkor a

LASTDRIVE = N

konfigurációs parancsot kell a CONFIG.SYS-ben elhelyezni. Ezután az A-N lemezazonosítókat használhatjuk. (Természetesen csak azokat, amelyekhez hozzá is rendeltünk egységet.)

11.2.8 SHELL - felhasználói parancs-processzor

A parancs lehetővé teszi, hogy a felhasználó saját parancs-processzorát töltse be a DOS rendszer indítása során.

Verzió: 2.11 - 3.30

A parancs formája:

SHELL = [*d:*][*út*]*file-név*[*.kit*][*paraméter1*][*paraméter2*]

A paraméterek jelentése:

Olyan file-t adunk meg, amely a DOS betöltése után a COMMAND.COM funkcióját veszi át.

Ily módon lehetőség van a parancs-processzor lecserélésére. A paraméterek jelentése a használt parancs-feldolgozótól függ.

11.2.9 STACKS - a DOS veremhasználatának módosítása

A konfigurációs parancs segítségével módosíthatjuk, hogy a DOS hány és mekkora vermet allokáljon egy-egy hardver megszakítás esetén.

Verzió: 3.20-3.30

A parancs formája:

STACKS = *n,s*

A paraméterek jelentése:

n a használható vermek száma. Alapértelmezése 9. Ezt az értéket 8 és 64 közé állíthatjuk be.

s az egyes vermek nagysága byte-ban. Alapértelmezése 128. Ezt az értéket 32 és 512 közé állíthatjuk be.

Megjegyzés:

A DOS minden egyes hardver megszakítas esetén allokál magának egy vermet a fenti paraméterek alapján kialakított vermekből. A megszakítás lekezelése után a verem területe ismét felszabadul.

12. A DOS speciális lehetőségei

12.1 A standard I/O átirányítása

Valamennyi DOS parancs három perifériát használ:

- standard input;
- standard output;
- standard periféria a hibaüzenetek kiírására.

Az egyes parancsok - feladattól függően - további perifériákat használhatnak. A DOS lehetőséget biztosít a standard input és output átirányítására egy másik perifériára (pl. az output-ot a nyomtatóra, vagy egy lemezes file-ba). A hibaüzenetek mindig a képernyőn jelennek meg.

Az output átirányításához a DOS parancs után az alábbiak egyikét kell írni:

a) *>periféria*
b) *>>periféria*

A periféria vagy egy logikai periféria neve, vagy egy file leírása. Ebben az esetben

[d:][út]file-név[.kit]

alakúnak kell lennie. Az a) esetben, ha lemezes file-ról van szó, a DOS egy új file-t hoz létre. A b) esetben, ha már létezik a file, a DOS standard outputja hozzáíródik a megfelelő file tartalmához.

Megjegyzés:

1. Az a) esetben, ha már létező file-t adunk meg, akkor az output a file első rekordjába kerül tehát felülírja, törli a régi adatokat.
2. A b) esetben, ha a file még nem létezik, akkor a rendszer azt létrehozza.

Az input átirányítására az alábbit kell a DOS parancs után írni:

<periféria vagy

<[d:][út]file-név[.kit]

A program vagy parancs a megadott file-ról vagy perifériáról várja az inputot.

Megjegyzés:

1. Ha az átirányított standard input file-ban valamilyen hiba fordul elő (pl. helytelen paraméterezés), akkor a parancs befejezi működését, és a DOS prompt jelenik meg a képernyőn. (A standard input periféria ismét a billentyűzet.)
2. Standard input átirányítása esetén a teljes inputot a file-nak kell tartalmaznia. Amennyiben a program végigolvasta a standard inputot, ezután a DOS onnan már csak "file-vége" jelet olvas be. Amennyiben a programnak még valamire szüksége

van, soha nem kapja meg, és könnyen lemerevedhet. Ilyenkor csak a <Ctrl-Break>-kel szakítható meg.

Példa:

1. A következő parancs megadása esetén az aktuális lemezegység tartalomjegyzéke nyomtatóra kerül:

```
DIR >PRN
```

2. Ebben az esetben pedig a DIR outputja egy már létező file-hoz hozzáíródik:

```
DIR >>FIXLEMEZ.DIR
```

3. Ebben a példában a standard input átirányítását mutatjuk be. Tegyük fel, hogy a PROGRAM1.EXE a billentyűzetről olvasná be az adatait. Ezeket az adatokat mi már egy file-ban: (PROGRAM.ADA) elhelyeztük. (Ezt a sorszerkesztő (EDLIN) vagy a COPY parancs segítségével is megtehetjük.) Ezután a program indítása a következő módon is történhet:

```
PROGRAM1<PROGRAM.ADA
```

12.2 A standard I/O-k összekapcsolása

Ez a lehetőség biztosítja, hogy több program/parancs végrehajtását összekapcsoljuk úgy, hogy az egyik standard outputja a következő standard inputja legyen.

Ebben az esetben a DOS a főtartalomjegyzékben egy ideiglenes file-t alakít ki, aminek az azonosítója %PIPEx.$$$ lesz. A DOS gondoskodik arról, hogy az első parancs standard outputja a %PIPEx.$$$ file-ba kerüljön, s hogy a következő parancs onnan olvassa a standard inputot. Az esetleges hibaüzenetek a képernyőre kerülnek.

A standard I/O összekapcsolások lehetőségét "PIPE"-funkciónak vagy "PIPE" típusú parancsláncnak is nevezik.

Formája a következő: *parancs*[¦*parancs*]...

Az összekapcsolásra kerülő programok/parancsok azonosítóit a ¦ jellel kell elválasztani. Ha ez a jel a billentyűzeten nem található, akkor az <ALT> billentyű és az 1,2,4 numerikus billentyűk lenyomása (a klaviatúra jobb oldalán) pótolja.

Megjegyzés:

1. Az előző parancs/program outputja nem jelenik meg a képernyőn.

2. A %PIPEx.$$$ azonosítóban az x az ideiglenes file sorszáma.

3. A parancs1 ¦ parancs2 hatása azzal azonos, mintha egymás után kiadtuk volna a

```
parancs1 >%PIPEx$$$
parancs2 <%PIPEx.$$$
DEL %PIPEx.$$$
```

DOS parancsokat.

Példa:

1. A fejezet további részében ismertetésre kerül a SORT parancs. Ha a következő parancsot adjuk meg:

DIR ¦ SORT >RENDEZET.DIR

Ekkor a DIR outputja - ami az aktuális meghajtó tartalomjegyzéke - lesz a SORT inputja. Ezt a SORT rendezi, majd a SORT outputja a RENDEZET.DIR file-ba kerül. Később ez a TYPE paranccsal a képernyőre listázható.

12.3. Speciális DOS parancsok

12.3.1 MORE - lista oldalakra tagolás

A MORE ún. filter-parancs. Az adatokat a standard inputról olvassa és azt a standard outputra írja, de egy képernyőnyi adat átvitele után üzenetet küld, és szünetel a kijelzés. Ha lenyomunk egy tetszőleges billentyűt, akkor folytatódik az átvitel. Ez mindaddig ismétlődik, amíg valamennyi input adat átvitelre nem kerül, (illetve <Ctrl-Break>-kel meg nem szakítjuk a parancsot).

Verzió: 2.11 - 3.30
A parancs típusa: Külső
A parancs formája:

MORE <[*d:*][*út*]*file-név*[*.kit*]

Megjegyzés:

Az angol, illetve a német nyelvű rendszerekben a következő üzenetet kapjuk:

- MORE -, illetve - FORTSETZUNG -

Példa:

A PROG1.ASM program sorait írjuk a képernyőre. Ha a képernyő megtelt, akkor üzenetet küld a rendszer, majd egy billentyű lenyomásával folytatódik a kijelzés. Egyben példát is látunk a standard input átirányítására.

A parancs a következő:

MORE <PROG1.ASM

Amennyiben a MORE parancsot a standard input átirányítása nélkül hívtuk meg (pl. elfelejtettük a '<' jelet hívni), a program a billentyűzetről vár inputot. Vagy adjunk <F6>-ot, majd <ENTER>-t - ekkor a program terminál, vagy abortáltassuk <Ctrl-Break>-kel.

12.3.2 FIND - keresés, válogatás

A FIND ún. filter-parancs. A parancs paraméterében meghatározott file (esetleg több file) rekordjaiban egy meghatározott karaktersorozatot keres. Azok a rekordok, amelyek a megadott karaktersorozatot tartalmazzák, a standard outputon kijelzésre kerülnek.

Verzió: 2.11 - 3.30
A parancs típusa: Külső
A parancs formája:

FIND [/V][/C][/N]*karaktersorozat*[[*d:*][*út*]*file-név*[*.kit*]...]

A paraméterek jelentése:

/V paraméter megadása esetén minden olyan rekordot kijelez, amelyik a megadott karaktersorozatot tartalmazza

/C paraméter megadása esetén csak azoknak a rekordoknak a számát írja ki, amelyek a megadott karaktersorozatot tartalmazzák

/N alkalmazása esetén a rekord relatív sorszámát, majd a rekordot is kijelzi egyezés esetén.

Megjegyzés:

1. A karaktersorozatot idézőjelek (") között kell megadni.

2. A globális file-név jelek (?,*) nem használhatók a file-ok kijelölésére.

Példa:

A FIND egy összetett alkalmazására mutatunk példát. A standard I/O összekapcsolásával egy lemez tartalomjegyzékében a .COM karaktersorozatot keressük.

A > DIR C¦FIND/V".COM"

A parancssorozat valamennyi .COM kiterjesztésű file-nevet kiírja.

12.3.3 SORT - rendezés

A SORT ún. filter-parancs. A standard inputról adatokat olvas be, azokat rendezi, majd a standard outputra írja.

Verzió: 2.11 - 3.30
A parancs típusa: Külső
A parancs formája:

SORT [/R][/+*n*]

A paraméterek jelentése:

/R megadása esetén a rendezés alfabetikusan csökkenő lesz. Az alapértelmezés a növekvő sorrend. (Reverse - fordított: a B karakter az A előtt áll.)

/+*n* helyén egy nem negatív egész szám áll. Ez azt jelöli, hogy a rekord n-edik karakterétől (poziciójától, oszlopától) történik a rendezés. Ha nem adunk meg értéket, akkor 1 az alapértelmezés, tehát a rekord első karakterétől történik a rendezés.

Példa:

A SORT egy összetett alkalmazására mutatunk példát. A standard I/O összekapcsolásával a tartalomjegyzék rekordjait a file-ok mérete szerint rendezzük és ezt nyomtatóra átirá-

nyítjuk:

```
DIR C: ¦SORT /+14 >PRN.
```

Ha az outputot egy file-ba szeretnénk átirányítani, akkor ez a következő parancsokkal történhet:

```
DIR C: ¦SORT/+14>RENDEZET.DIR
```

Megjegyzés:

1. A rendezendő file mérete legfeljebb 63 Kbyte lehet.
2. A tabulátor jeleket a SORT nem üres karakterként értelmezi.
3. A karaktereket a SORT bináris értékük szerint rendezi, kivéve a következő eseteket:

- a kisbetűket a-tól z-ig ugyanolyan súllyal veszi a rendezésnél figyelembe, mint a nagybetűket A-tól Z-ig, azok egyenlő eséllyel kerülnek - tehát előfordulási sorrendben - ugyanarra a helyre;
- a decimális 127-nél nagyobb ASCII értékű karaktereket a következő táblázat szerint veszi figyelembe:

ASCII érték (dec)	ASCII rendezési súly	ASCII érték (dec)	ASCII rendezési súly
128	67	153	79
129	85	154	85
130	69	155	36
131	65	156	36
132	65	157	36
133	65	158	36
134	65	159	36
135	67	160	65
136	69	161	73
137	69	162	79
138	69	163	85
139	73	164	78
140	73	165	78
141	73	166	166
142	65	167	167
143	65	168	63
144	69	169	169
145	65	170	170
146	65	171	171
147	79	172	172
148	79	173	33
149	79	174	34
150	85	175	34
151	85	225	83
152	89		

12.3.4 PROMPT - a prompt-karaktersor megváltoztatása

A parancs segítségével a képernyőn megjelenő prompt-karaktert, vagy karaktersorozatot (rendszerazonosítót) lehet beállítani.

Verzió: 2.11 - 3.30
A parancs típusa: Belső
A parancs formája:

PROMPT [*szöveg*]

A "*szöveg*" lesz az új rendszer prompt szövegrésze. Ebben a szövegben előfordulnak speciális, két karakterből álló karaktersorozatok is. E karaktersorozat első karaktere a **$**. A második karakter a következő értékeket veheti fel:

$: a dollárjel
t: idő
d: dátum
p: aktuális tartalomjegyzék az aktuális meghajtón
v: a DOS verziója
n: az aktuális meghajtó betűjele
g: a > karakter
l: a < karakter
b: a ¦ karakter
q: az = karakter
h: a kurzor visszaléptetését és a megelőző karakter törlését jelöli
e: az ESCape karakter
_: a CR/LF (Carrige Return - Line Feed; kocsi vissza-soremelés) a képernyő következő sorának az elejére kerül a kurzor.

Egyéb értékek megadását a DOS figyelmen kívül hagyja. Ha a PROMPT parancsot "*szöveg*" nélkül adjuk meg, visszaáll az eredeti (azaz a "C>" alakú) DOS prompt.

Megjegyzés:

1. Ha a $ karakter maga is szerepel a "szöveg" karaktersorozatban, akkor azt **$$** formában kell megadni.

2. A $e (ASCII-kód X"1B") megadásával escape-sorozatok állíthatók elő. Ezeket a kibővített képernyő- és billentyűzet-kezelés című alfejezetben ismertetjük.

Példa:

1. Példa olyan "szöveg"-re, amely szintén a normál DOS promptot állítja vissza:

PROMPT ng

2. PROMPT 123ABC hatására a rendszer promptja az 123ABC szöveg lesz.

3. Az új prompt karaktersorozat a rendszer dátum és a rendszer idő lesz, ha a következőt adjuk meg:

PROMPT d_$t

A dátum és az idő nem egy sorban kerül kijelzésre.

4. Ebben a példában a prompt karaktersorozat az aktuális meghajtó azonosítója lesz, de az escape-szekvencia megadásával a kijelzés "fordított" módon (reverse video) történik. A parancs a következő:

PROMPT $e[7m$n:$e[m

$e[7m: az inverz video kijelzés bekapcsolása,
$e[m: a normál kijelzés visszaállítása a képernyőn.

12.3.5 CTTY - alternatív konzol kijelölése

A parancs segítségével az aktuális input/output konzol egy alternatív konzolra változtatható, illetve az aktuális billentyűzetet és a képernyőt ismét konzolként használja a DOS.

Verzió: 2.11 - 3.30
A parancs típusa: Belső
A parancs formája:

CTTY *egységazonosító*

A paraméter jelentése:

Az *egységazonosító* a következő lehet (a kettőspont használata nem kötelező):

AUX: az első aszinkron adapter,
COM1: az első aszinkron adapter,
COM2: a második aszinkron adapter,
COM3: a harmadik aszinkron adapter, (DOS 3.30-tól)
COM4: a negyedik aszinkron adapter, (DOS 3.30-tól)
CON: az aktuális billentyűzet, képernyő.

Megjegyzés:

1. A CTTY bármilyen karakteres perifériaazonosítót elfogad, így például a PRN-t, a nyomtatót is. Ez a rendszer leállásához vezet, mivel a nyomtató nem tudja végrehajtani a rendszer által kiadott input parancsokat (például: olvasási parancs).
2. A felhasználó saját karakteres perifériájának az azonosítóját is megadhatja, ha az input és output műveleteket is végre tud hajtani.

Példa:

1. A DOS az AUX egységet tekinti az aktuális konzolnak, ha a következő parancsot kiadjuk:

CTTY AUX

2. Az aktuális klaviatúra, képernyő eredeti hozzárendelés visszaállítása a következő paranccsal történik:

CTTY CON

12.3.6 EXE2BIN - átalakítás .EXE-ről .COM formátumra

A parancs lehetővé teszi, hogy a LINK programszerkesztővel készített .EXE kiterjesztésű programot átalakítsuk .COM formátumra. A .COM formátumú programot a parancsprocesszor parancsként hajtja végre.

Az átalakítás következtében a program mérete csökken és betöltése is gyorsabb lesz.

A .COM és az .EXE formátumról a 15. fejezetben (a PC DOS munkaterületei és vezérlő blokkjai) találhatók további információk.

Verzió: 2.11 - 3.30
A parancs típusa: Külső
A parancs formája:

EXE2BIN [*d:*][*út*]*file-név*[*.kit*] [*d:*][*út*][*file-név*[*.kit*]]

A paraméterek jelentése:

Az első paraméterrel az EXE formájú input programot határozzuk meg. A file-név kiterjesztésének az alapértelmezése: .EXE. A második paramétert csak akkor kell megadni, ha a COM formájú program neve változik vagy a nevének a kiterjesztése nem .COM lesz.

Megjegyzés:

1. Ha az első file nem .EXE formájú, akkor a következő üzenetet kapjuk:

   File cannot be converted, illetve
   Datei kann nicht umgestellt werden.

2. Az átalakítás nem végezhető el, ha az átalakítandó program 64 Kbyte-nál nagyobb méretű, illetőleg, ha a betöltési szegmenstől függő relokációs elemet tartalmaz (ilyen a STACK szegmens is!).

3. A .COM formátum az assembler nyelvű programok esetén az ORG assembler direktíva segítségével alakítható ki:

```
ORG 100H
START:
.
.
.
END START
```

12.3.7 SET - programkörnyezet kezelése

Minden végrehajtásra kerülő program előtt a memóriában egy lefoglalt tárterület helyez-

kedik el, amely tartalmazza a DOS aktuális paramétereit. Ez a terület definiálja a programkörnyezetet. A SET parancs lehetőséget ad a programkörnyezet lekérdezésére, átírására, tartalmának bővítésére. A programkörnyezet részletes ismertetése megtalálható a "PC-DOS munkaterületei és vezérlőblokkjai" című fejezetben.

Verzió: 2.11 - 3.30
A parancs típusa: Belső
A parancs formája:

SET [*kulcsszó* = [*paraméterérték*]]

A parancs megadható formái:

- A SET parancs önmagában áll, sem kulcsszó sem paraméterérték nem követi. Ilyen esetekben a képernyőre kiíródik a programkörnyezet valamennyi kulcsszava az aktuális paraméterértékekkel együtt. Pl.:

  A > SET ennek hatására kiíródik:
  COMSPEC = C:\
  PROMPT = PG
  PATH = \;\SYSTEM

- A SET parancsot egy kulcsszó és egyenlőségjel követi: "SET prompt = ". Ennek hatására a parancsban megadott kulcsszó és a hozzá tartozó paraméter törlődik a programkörnyezetből. Ezután a "SET" már csak az alábbi kiírást eredményezné a képernyőn:

  COMSPEC = C:\
  PATH = \;\SYSTEM

- A SET parancsban a kulcsszót követő egyenlőségjel után paramétert is megadunk. Ebben az esetben az így megadott kulcsszó és paraméter bekerül a programkörnyezetbe. Ha kulcsszó már szerepel benne, akkor a korábbi megadás törlődik. Például:

  SET printscrn = yes

parancs hatására a programkörnyezetbe bekerül a "PRINTSCRN = yes" jelsorozat. Ha történetesen van olyan felhasználói program, amely abban az esetben, ha a PRINTSCRN értéke yes (vagy YES), minden üzenetet képernyőre és printerre is kiküld, különben csak képernyőre, úgy ez a program ezután printelni is fogja az üzeneteit.

Megjegyzés:

1. Amint az utolsó példában is látható, a PC-DOS a kulcsszót mindig nagybetűsre konvertálja, továbbá igy hasonlítja és tárolja.

2. A PC-DOS csak az első példában is szereplő COMSPEC, PROMPT és PATH kulcsszavakat értelmezi. Ezek közül a COMSPEC a rendszer konfigurálásakor állítható. A másik kettő önálló parancsként is megadható. A kétféle megadás teljesen egyenértékű és azonos hatású.

3. A programkörnyezet mérete legfeljebb 32 Kbyte lehet.
4. Az érvényes programkörnyezet-szegmens címe a betöltött program PSP-jének (Program Segmens Prefix) hexadecimális 2C relatív címén található.
5. A rezidens programok környezetleírását már sem a SET, sem a PROMPT, sem a PATH parancs nem változtatja meg.
6. A környezeti változók a parancs-file-okban használhatók (lásd 7.1.2!).

12.3.8 NLSFUNC - ország-függő információk kezelése

Az NLSFUNC parancs segítségével lehetőségünk nyílik a kibővített ország-függő információk kezelésére. A program egy része a memóriában marad, annak segítségével látja majd el a DOS ezta funkciót.

Verzió: 3.30
A parancs típusa: külső
A parancs formája:

NLSFUNC [[*d:*][*út*]*file-név*.[*kit*]]

A paraméter jelentése:

Az NLSFUNC után szereplő file-név az ország-függő információkat tartalmazó file-t azonosítja. Ha a paramétert nem adjuk meg, akkor a CONFIG.SYS file-ban elhelyezett COUNTRY= parancsban szereplő file-t használja. Ha ott sincs megadva, akkor a COUNTRY.SYS file-t keresi.

12.3.9 CHCP - kódlap váltás valamennyi eszközre

A parancs segítségével megadhatjuk, hogy a DOS melyik kódlappal dolgozzon. A DOS valamennyi eszköze módosítja a kódlapot, feltéve, hogy ezt valamilyen formában (pl. egy MODE CP PREP paranccsal) megengedtük.

Verzió: 3.30
A parancs típusa: külső
A parancs formája:

CHCP [*kp*]

A paraméter jelentése:

kp a kívánt kódlap sorszáma.

Megjegyzés:

A CHCP kiadása előtt le kell futtatni az NLSFUNC programot is.

EltSECRET

EltSECRET
adatvédelmi rendszer

Termékünk, az EltSECRET, a SW'88 díjazott programjának, az EltGUARD-nak adatvédelemre továbbfejlesztett változata.

IBM PC-n működő adatvédelmi rendszer, mely nemcsak az adatokhoz és programokhoz való hozzáférést szabályozza, hanem adattitkosításra (kódolásra) is lehetőséget ad.

Az EltSECRET rendszer jellemzői:

- 64 hierarchia-szint a fájlok védelmére
- 127 alkönyvtárszint
- 2500 jogosult felhasználó
- 3 különböző bonyolultságú kódolási eljárás
- együttműködés lokális hálózati rendszerekkel
- felhasználói menürendszerek kialakítása
- kiegészítő hardverrel fokozott biztonság

Hardverkiegészítő használata esetén a felhasználók fizikai személyi azonosítót kapnak, jelszavuk mellett ez is kell a bejelentkezéshez. A herdver letiltja a floppyról történő rendszerbetöltést is.

A teljes program ára: **76.000,– Ft** (+ ÁFA).
A DEMO változat ára: **8.000,– Ft** (+ ÁFA).

Egy telephely összes számítógépéhez elegendő egy példány EltSECRET rendszert vásárolni.

Bővebb felvilágosítással, bemutatóval állunk rendelkezésére:

eltsoft

EltSOFT
Számítástechnikai és Reklámszervező GMK
1122 Budapest, Városmajor u. 7–9.
Telefon: 550-962 1536 Budapest, Pf. 390.

Kedves Olvasó!
Tisztelt DOS alkalmazó!

Tudja-e, ...

- hogy az OTP 8 fővárosi, valamint 2 vidéki fiókjában a csekkforgalmat a mi hálózati csekkszámla-rendszerünk segítségével bonyolítják...
- hogy a HUNGÁRIA BIZTOSÍTÓ országos hálózatában (33 vidéki és 4 fővárosi fióknál) a gépjármű-kárszámítást támogató rendszert mi fejlesztettük...
- hogy országos szinten a megyei és helyi tanácsoknál 9 tanácsi munkát támogató rendszerünknek 201 felhasználója van...
- hogy számos iparvállalatnál, gazdálkodó egységben üzemelnek a vállalati ügyvitelt és irányítást támogató rendszereink...
- hogy felhasználói rendszereink (hálózati alkalmazásaink) túlnyomó többségét a REXLIB fejlesztőrendszerrel fejlesztettük, mely saját szoftvertermékünk...
- hogy nemcsak saját fejlesztésű rendszereinkhez forgalmazunk IBM PC XT/AT kompatibilis számítógépeket a kívánt kiépítésben...
- hogy **Önt is szívesen látjuk partnereink között!**

SZÁMSZÖV®
SZÁMÍTÁSTECHNIKAI KISSZÖVETKEZET

1116 Budapest XI., Hunyadi J. u. 162
Levélcím: 1430 Budapest, Pf. 16.
Telefon: Titkárság: 667-809
Központ: 665-322

Dél-dunántúli iroda: 7632 Pécs, Sarohin tábornok u. 31. VII. 25.
Levélcím: 7616 Pécs 16, Pf. 4.
Telefon: (72)23-869 üzenetrögzítővel

COMPUTERSHOP üzlet: Budapest, Népszínház u. 37.
Telefon: 336-285